Numerical Methods for Real Options in Telecommunications

by

Yann d'Halluin

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2004

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis applies modern financial option valuation methods to the problem of telecommunication network capacity investment decision timing. In particular, given a cluster of base stations (wireless network with a certain traffic capacity per base station), the objective of this thesis is to determine when it is optimal to increase capacity to each of the base stations of the cluster.

Based on several time series taken from the wireless and bandwidth industry, it is argued that capacity usage is the major uncertain component in telecommunications. It is found that price has low volatility when compared to capacity usage. A real options approach is then applied to derive a two dimensional partial integro-differential equation (PIDE) to value investments in telecommunication infrastructure when capacity usage is uncertain and has temporary sudden large variations.

This real options PIDE presents several numerical challenges. First, the integral term must be solved accurately and quickly enough such that the general PIDE solution is reasonably accurate. To deal with the integral term, an implicit method is suggested. Proofs of timestepping stability and convergence of a fixed point iteration scheme are presented. The correlation integral is computed using a fast Fourier transform (FFT) method. Techniques are developed to avoid wrap-around effects. This method is tested on option pricing problems where the underlying asset follows a jump diffusion process.

Second, the absence of diffusion in one direction of the two dimensional PIDE creates numerical challenges regarding accuracy and timestep selection. A semi-Lagrangian method is presented to alleviate these issues. At each timestep, a set of one dimensional PIDEs is solved and the solution of each PIDE is updated using semi-Lagrangian timestepping. Crank-Nicolson and second order backward differencing timestepping schemes are studied. Monotonicity and stability results are derived. This method is tested on continuously observed Asian options.

Finally, a five factor algorithm that captures many of the constraints of the wireless network capacity investment decision timing problem is developed. The upgrade decision for different upgrade decision intervals (e.g. monthly, quarterly, etc.) is studied, and the effect of a safety level (i.e. the maximum allowed capacity used in practice on a daily basis—which differs from the theoretical maximum) is investigated.

Acknowledgements

This thesis is the result of four years of work whereby I have been supported and helped by many people. It is now my turn to express my thanks to all of them.

First off, I would like to thank Bell University Laboratories at the University of Waterloo for funding this work. In particular, I would like to express my gratitude to Gerry Kealey, Derek McAvoy and Darlene Farrow for many useful discussions.

I would like to thank Professor Peter Forsyth who has been my supervisor since 1998 when I started my MMath. During these years, Professor Forsyth has always been there to help and guide me in my research and his enthusiastic cry of "Any Progress yet" will haunt me for the rest of my days.

I would like to thank Professor Ken Vetzal for countless hours of discussions and all his contributions to my papers and my thesis. My "righting" would not have been what it is without his help.

I would also like to thank my committee members for taking the time to review my thesis and providing me with valuable comments: Leif Andersen, Ken Vetzal, George Labahn, and Phelim Boyle.

I would also like to express my gratitude to Professor George Labahn for taking the chance with me as a graduate student and for his encouragement.

Last but not least, I would like to express my gratitude and thanks to all the people that have made my time at Waterloo both productive and enjoyable. Members of the scicom lab, both past and present. Particular mention must be made of Reinhold, Dave, Howard, Rob, Simon, Ron, Shannon and Dana. Special appreciation goes to Amélie for her patience and help. My time at Waterloo will be cherished for all the good friends, good food and knowledge that I have acquired.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

As evidenced by the current over-capacity in long haul optical fibre networks, it is clear that current telecommunication industry investment decision mechanisms need to be re-evaluated. Traditional capital budgeting techniques such as net present value and internal rate of return are now recognized as being deficient in dealing with strategic management of corporate investments over time. These traditional methods do not take into account management's ability to respond to changing business conditions. Consequently, traditional methods can provide misleading estimates of the value of corporate investments under uncertainty. Moreover, these methods do not provide a clear indication of how management should respond to new conditions (e.g. when should a corporate investment project be mothballed or abandoned completely, or when should its scale be expanded).

The real options approach is an emerging alternative methodology which addresses these deficiencies. It uses techniques adapted from those developed for valuing and hedging financial derivative contracts (e.g. options, futures). In particular, optimal actions for management are determined using dynamic programming methods, and these in turn are used to calculate the value of a corporate investment project.

This thesis is directed toward demonstrating the use of real options decision making

tools for optimal capacity choice for network planning in telecommunications. Given a set of possible capacity upgrade levels, a telecommunication firm has to decide which level is optimal to upgrade to under uncertain factors. We aim to find the percentage (in terms of the maximum equipment transmission capacity) at which level it is optimal to upgrade.

To the best of our knowledge, this approach has not been used previously. Dixit and Pindyck [35] study the problem of incremental upgrade in the context of a multi stage project in the oil industry. Dangl [26] presents a model where a firm has only one chance to determine optimal investment timing and optimal capacity. Once the capacity is fixed the project output upper bound is fixed and cannot be adjusted later. Our approach can be viewed as a combination of both of these methods. The firm investing into new equipment can upgrade its network at any time, independently of the equipment lead time, and different levels of capacity are reachable. Based on a real options approach a partial integro-differential equation (PIDE) for the investment decision timing problem is derived.

1.1 Contributions

The main contributions of this thesis are as follows. In Chapter 2 we show that the main underlying risk factor when investing in telecommunications is capacity usage. Capacity usage is very volatile, while the spot price paid by customers per unit of traffic has very low volatility. Based on these observations and some empirical studies, a two factor stochastic process is presented to model telecom traffic. Using a real options approach a partial integro-differential equation (PIDE) is then derived for the investment value. The real options PIDE has two features which make numerical solution challenging: an integral term, and convection dominance. We examine these two numerical issues separately.

In Chapter 3 we show that:

- the jump diffusion term of an option pricing PIDE can be discretized explicitly, and

when coupled with a fully implicit treatment of the usual PDE, the resulting timestepping method is unconditionally stable.

- if an implicit timestepping method is used, the solution of a dense matrix can be avoided by using a simple fixed point iteration scheme to solve the discretized algebraic equations. This iteration is shown to be globally convergent.

- it is possible to develop a method for efficiently computing the jump integral term using FFT methods. No assumptions are made about the probability density for the jump term, except that the jump size density has finite expectation. This general approach requires the evaluation of correlation type integrals, as in [92]. We also show how to eliminate the wrap-around effects which often plague FFT methods.

- in contrast with previous work, the grid is not assumed to be equally spaced in either the underlying asset price or its logarithm. This is a major advantage for the pricing of contracts with barrier provisions, which typically require fine grid spacing near barriers in order to achieve sufficient accuracy.

- the discrete penalized equations which result from the discretization of the differential-integral complementarity problem for pricing American options on assets which follow a jump diffusion process, can also be solved using a fixed point iteration. Sufficient conditions at each timestep for the global convergence of this iteration are also derived.

- the method developed here uses implicit timestepping for both the correlation integral term and the American constraint. As a result, higher order convergence (in terms of timestepping error) is obtained when compared with previous methods which treat the correlation integral or the American constraint explicitly.

In Chapter 4 we show that:

- using the methods developed in Chapter 3, quadratic convergence is obtained when pricing a wide variety of financial options. Even if the initial conditions are not smooth, such as for digital options or Parisian options, quadratic convergence was recovered.

- for typical values of the timestep size and Poisson arrival intensity, the fixed point iteration scheme used to handle the jump integral term converges very rapidly (the error is reduced by two orders of magnitude at each iteration). Given typical market parameters [5], three iterations per timestep are enough for the fixed point algorithm to achieve convergence to a reasonable tolerance.

- when pricing American options under the jump diffusion process, the number of iterations per timestep is relatively unchanged compared to pricing vanilla options under the jump diffusion process.

In Chapter 5 we show that:

- a semi-Lagrangian method can be used to price continuously observed Asian options (which is a prototypical example of convection dominance), where the underlying asset is assumed to follow a jump diffusion process. The implementation suggested here reduces this problem to solving a decoupled set of one dimensional discrete partial integro-differential equations at each timestep. This makes implementation of this method very straightforward in a software library which is capable of pricing discretely observed path dependent options [95].

- in the fully implicit case, the semi-Lagrangian method is algebraically identical to a standard numerical method for pricing discretely observed Asian options, when the observation interval is equal to the discrete timestep [94].

- since the discretized problem at each timestep reduces to a set of decoupled 1d PIDEs,

we can make use of the techniques developed in Chapter 3 to prove certain properties of the discrete scheme, including convergence of the iterative method used to solve the implicit discrete equations. In the fully implicit case, it is straightforward to prove l_∞ stability.

- even if second order timestepping methods are used, convergence as the mesh and timestep is refined occurs at a less than second order rate. The problem can be traced to the non-smoothness of the payoff function. The SL approach contrasts with the method traditionally used in industry (see [95]) where high order timestepping schemes are used but the jump condition is applied explicitly resulting only in linear convergence.

In Chapter 6 we show that:

- the methods developed for path dependent options, where the underlying follows a jump diffusion process, can be applied to the telecom investment problem. The limitations of the algorithm presented in [31] are eliminated by solving a five factor path dependent problem. This enables us to consider different upgrade decision intervals independent of equipment lead time.

- today's upgrade decision, in terms of the optimal upgrade percentage, increases as the value of the market price of risk increases.

- when considering jumps in capacity usage, the volatility and the growth rate also change which affects today's upgrade decision. It is found that jumps in capacity usage delay the decision to upgrade.

- by developing appropriate penalty functions which assign a cost to poor quality of service, we can combine both financial and quality of service criteria. This approach requires managers to assign a cost to quality of service issues. Penalty functions could

be real financial incentives provided to users (e.g. during high blocking periods, all calls are free), or they could be based on customer churn rates.

1.2 Outline

The remainder of this thesis is organized as follows. Chapter 2 presents a two dimensional partial integro-differential (PIDE) equation to value investment into telecommunication equipment. This equation is the foundation of the optimal decision timing investment problem we are solving in this thesis. However, before developing algorithms, several numerical issues must be addressed.

The two factor investment value PIDE has an integral term that must be handled carefully. Chapter 3 proposes a method for handling the jump integral term in an implicit fashion. To validate this approach, a one dimensional PIDE taken from the financial derivative pricing literature is considered. Chapter 4 presents several numerical tests of convergence for a wide variety of financial options.

Chapter 5 addresses the issue raised by the absence of diffusion in one direction of the two factor investment value PIDE by using a semi-Lagrangian method. This method is tested on the two dimensional PIDE used to price continuously observed Asian options. This problem was selected since the convective term becomes infinite as time moves closer to today; making this is an extremely difficult problem.

Finally, Chapter 6 applies modern financial option valuation methods to the problem of wireless network capacity investment decision timing. In particular, given a cluster of base stations (with a certain capacity per base station), we determine when it is optimal to increase capacity for each of the base stations contained in the cluster. Based on the real options formulation of Chapter 2, a five factor algorithm that captures many of the constraints of wireless network management is developed. This optimal upgrade algorithm

maximizes the value of the investment to the network operator. Conclusions are finally outlined in Chapter 7.

Chapter 2

The Investment Problem

In this Chapter, the fundamental risk factor in telecommunication equipment investments is identified. Based on several time series taken from the wireless and bandwidth industries, it is argued that capacity usage is the major uncertain component in telecommunications. Although revenue to the owner of a network is the prevailing price times the usage, it is argued that price has low volatility compared to usage. Consequently, as a first approximation, we can consider price to be deterministic. Using statistical tests, a stochastic process is then suggested for modelling capacity usage. Finally, a real options approach is applied to derive the partial integro-differential equation (PIDE) to value investments in telecommunication infrastructure.

2.1 Background

The value of any corporation is primarily determined by the quality of the investment decisions made by that firm. Suboptimal decisions can damage a business, sometimes irreparably. In many cases, suboptimal decisions are made because the right information is not collected or simply because there is not enough data. However, in some cases the fault

lies in the method used to make that decision.

Net Present Value (NPV) and Internal Rate of Return (IRR) are the traditional methods used to quantify the value of an investment opportunity [35, 85]. For example, the NPV of a project is the difference between how much the project is worth (in terms of the discounted future cash flows it is expected to generate) and the capital expenditure it requires. If the difference between the two is greater than zero, then an investment should be made. Otherwise, the investment should not be made. Closely related to the NPV, the IRR on a project is the required return that results in a zero NPV when it is used as the discount rate.

However, it is now widely recognized that NPV or IRR do not capture the optionality embedded in investment projects. The NPV approach, as traditionally applied, fails to sufficiently account for much of the optionality inherent in corporate investments. The reason is that a firm with an opportunity to invest is in effect holding an option analogous to a financial American call option [52]. For example, a company often has the option to abandon a project or expand if the output exceeds expectations. Another problem with traditional NPV lies in the estimation of the appropriate risk-adjusted discount rate.

As an alternative, the real options approach [35, 85] can be used to effectively model investment flexibility. Real options theory does not render the traditional NPV methods obsolete, but it does help to capture the flexibility often present in an investment. Note that, provided the values of all options are appropriately included in the analysis of a project investment, the NPV approach will give the same conclusions as the real options approach.

This thesis studies the problem of optimal telecommunication investment decisions. Wireless telephones, video conferencing and email are now regarded as essential communication tools, dramatically impacting how people approach personal and business communications. As new network infrastructure is built and competition between companies increases, subscribers are becoming ever more critical of the service and voice quality they receive from

network providers. Network operators must provide a guaranteed level of service to customers while maximizing profit. The current environment, with decreasing revenue per unit of traffic and increasing demand on networks from new features in equipment, places strain on network managers.

While a number of publications discuss the use of real options theory for optimal investment timing (e.g. [35, 85] and references therein), very little work has been done in telecommunications planning and management [26].

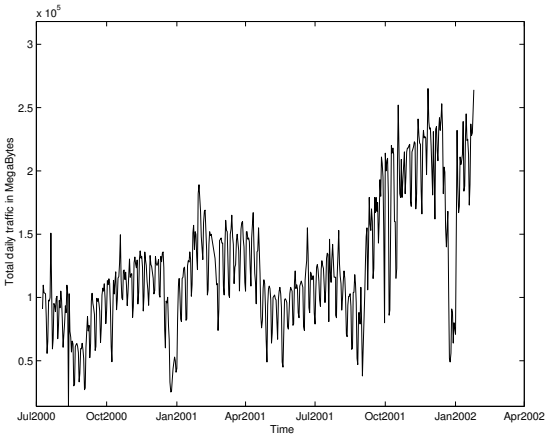
2.1.1 A Question of Demand

In order to use the real options approach to analyze investment decisions in telecommunications, the principal factors that are likely to affect the profitability of these investments must be identified.

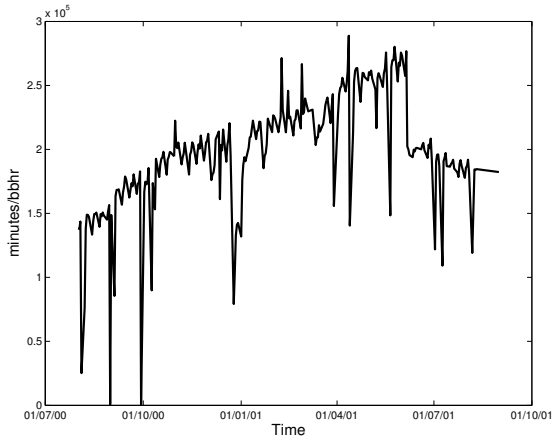
As we shall see, in the telecommunications industry, the fundamental uncertain factor driving profitability is the amount which can be sold, as opposed to the price received per unit. This situation is in contrast with traditional financial markets where the price is the dominating factor. For example, the owner of an optical fiber network faces this type of situation. New wavelength services allow the user to purchase limited capacity for days or even hours [47]. Effectively, users pay only for the bandwidth that they use. Similarly, revenues to the owner of a wireless network are determined by the prevailing price per minute and the amount of traffic. Consequently, the revenue to the owner of the network is determined by the prevailing price and the amount used.

A study of both bandwidth and wireless traffic data revealed some interesting facts. Capacity usage is highly volatile. Although traffic has increased, it has done so in an extremely erratic fashion (see Figure 2.1). Our estimates of volatility for bandwidth usage are in the range of 80% to 150% per year, and volatility for wireless usage is in the range of 50% to 90% per year [31, 33]. This can be compared with a volatility of 15% to 30% per

year for most major stock indexes.



(A) *Total daily Internet traffic into and out of the University of Waterloo from July 2000 to April 2002.*



(B) *Total daily busy hour minutes traffic for a wireless switch. Data is collected during the bouncing busy hour. This is the hour of the day with the most traffic.*

FIGURE 2.1: *Time series capacity usage examples.*

By comparison, price has shown a downward trend with relatively low volatility. Average revenue per wireless user is decreasing with relatively little uncertainty (Figure 2.2). Similarly, bandwidth prices have systematically decreased over the last few years (Table 2.1). As a result, in our opinion capacity usage is the main risk factor for telecommunication investments.

Table 2.1 Prices for lines with different maximum transmission rates. Prices are in \$/year/DS-0 mile. A DS-0 line has a maximum transmission rate of 64 kbps.

	Q1 1999	Q1 2000	Q1 2001	Q1 2002
OC-3	0.013	0.011	0.0082	0.0055
OC-12	0.012	0.01	0.0066	0.0045
OC-48	0.01	0.0095	0.0054	0.0035

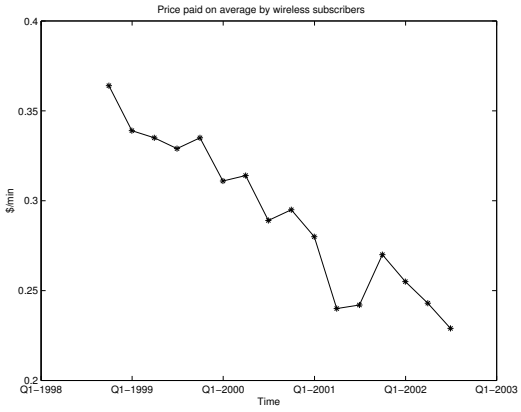


FIGURE 2.2: *Price paid on average by wireless network subscribers in dollars per minute. The price is obtained by dividing the average revenue per user by the usage per user. These data were obtained from Bell Canada quarterly financial reports.*

2.2 Modelling Capacity Usage

Capacity usage has obviously increased but this growth has been rather erratic (see Figure 2.1). While it would be possible to identify other uncertain factors such as, for example, technology advances or price per unit of traffic, in this thesis we consider only a single risk factor, the capacity used. Depending on the benefits of the approach presented here and its recognition by the telecommunication industry, it may be possible to consider in the future more complex models.

Any variable whose value changes over time in an uncertain way is said to follow a stochastic process [52]. As a first approximation, our basic assumption is that capacity usage Q can be modeled by a geometric Brownian motion process [52] defined by

$$dQ = \mu Q dt + \sigma Q dZ, \tag{2.1}$$

where μ is the *drift rate* or *growth rate* [89], σ is the *volatility* and dZ is the increment of a

Gauss-Wiener process [52, 89]. The Gauss-Wiener process increment can be written as

$$dZ = \phi\sqrt{dt}, \tag{2.2}$$

where ϕ is a random variable drawn from a normal distribution with zero mean and unit variance. The degree of randomness is determined by the volatility σ. In effect, when modelling capacity usage using equation (2.1), we are assuming that the logarithmic relative traffic changes are normally distributed [52]. In the absence of randomness ($\sigma = 0$), equation (2.1) has the solution

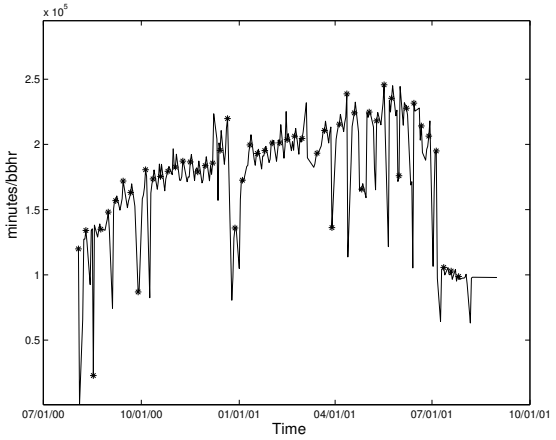
$$Q = Q_0 \exp\left(\int_0^t \mu(u)du\right), \tag{2.3}$$

where Q_0 is a constant.

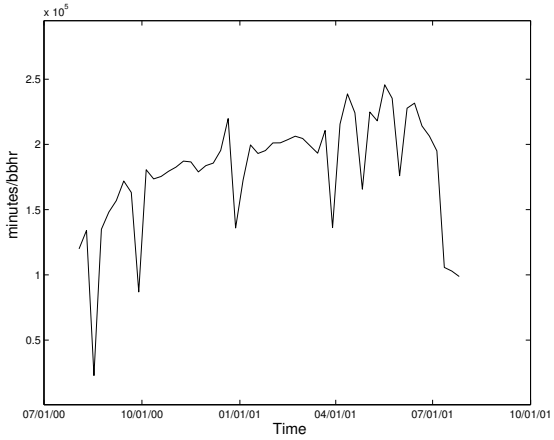
Before proceeding any further, the validity of equation (2.1) must be demonstrated. It is beyond the scope of this thesis to present all the different telecom usage time series that were made available to us. As such, equation (2.1) will be validated by focusing on a single time series taken from the wireless industry. For a more extensive analysis of bandwidth and wireless capacity usage data, readers are referred to [31] and [33]. Note that the model validation results presented in both of these papers are the same as the one outlined below.

An initial analysis of the capacity usage showed strong autocorrelation of the time series within each week (see Figure 2.3 (A)). This is not surprising, since we expect that there will be repetitive patterns within each week. To filter out this effect, network capacity usage is averaged for the entire year for each day of the week separately, and the day with the highest average total daily capacity usage is chosen. This same day each week is then chosen to estimate week to week effects (see Figure 2.3 (B)).

Prior to estimating the drift term μ and the volatility σ, the dates corresponding to vacation periods and sudden changes are smoothed out using interpolation. Major holidays are known events for low network capacity usage and should not be considered when com-



(A) *The stars indicate the weekly highest traffic days.*

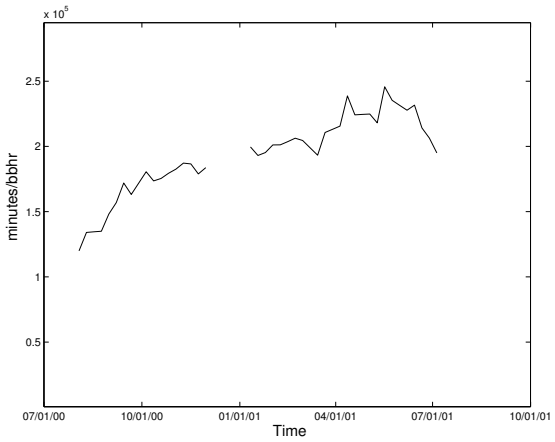


(B) *The data corresponds to the highest average day traffic per week. This day is chosen based on the average network traffic that day encounters throughout the year. Once that day is found, it remains the same for the rest of the year.*

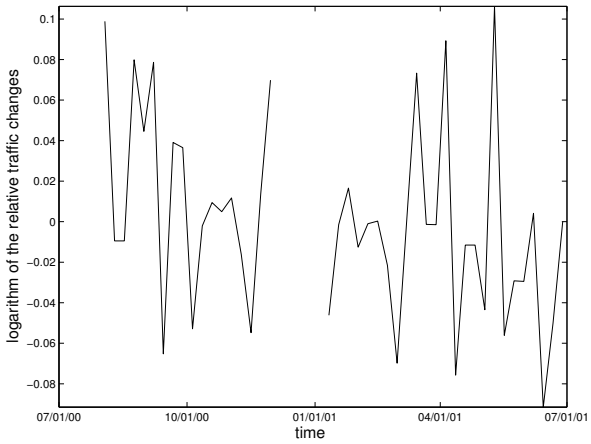
FIGURE 2.3: *One year wireless traffic data. This data is collected at daily intervals during one hour. This hour corresponds to the hour of the day with the most traffic. It is often referred to as the bouncing busy hour.*

puting the drift and the volatility. The major holidays are based on the Canadian calendar. An analysis of the data showed significant anomalous behavior on vacation dates. Studies also indicated that the month of December has lower traffic than the rest of the year. Consequently, the month of December is ignored since the decrease in traffic is a non-random event. Furthermore, by investigating graphically the different time series, it was possible to detect large drops in network traffic. These drops appeared across several time series and did not correspond to any major holidays. These were perhaps due to equipment failure (as suggested by our industrial contacts [64]). For now these changes will be smoothed out. They will be considered later (see Section 2.2.1).

Figure 2.4 presents the results once the holidays, and suspicious events, have been removed or smoothed out from the time series presented in Figure 2.3.



(A) *Vacation effects and suspicious changes have been suppressed using interpolation. The month of December is ignored.*



(B) *Logarithmic relative traffic changes. The trend of the time series has been removed*

FIGURE 2.4: *One year wireless traffic data. This data is collected at daily intervals during one hour. This hour corresponds to the hour of the day with the most traffic. It is often referred to as the bouncing busy hour in the industry.*

Table 2.2 Details of the Ljung-Box statistical test for the wireless time series. The data in the time series represents the minutes bouncing busy hour traffic. The Ljung-Box test is executed at the 5% significance level. H is a boolean decision variable. $H = 0$ indicates acceptance of the null hypothesis that the model fit is adequate (no serial correlation at the corresponding element of lags); $H = 1$ indicates rejection of the null hypothesis. P-values are the significance levels at which the null hypothesis of no serial correlation at each lag in Lags is rejected.

	Before the month of December				After the month of December			
lag	1	2	3	4	1	2	3	4
H	0	0	0	0	0	0	0	0
pvalue	0.62	0.78	0.73	0.83	0.78	0.93	0.85	0.92

If the stochastic process defined by equation (2.1) is a good model for capacity usage, then the detrended logarithmic relative traffic change time series should contain only random data (see Figure 2.4). To assess the hypothesis that data contains only white noise, the Ljung-Box $\mathcal{Q}$-statistical test is used [23]. Tests are conducted at the 5% significance level

for the period before and after December for different lags (e.g. lag = 1, 2, 3 and 4). For example, if both tests (before/after December) (see Figure 2.4) return 0 for lag equal to 4, then the data is random. If at least one of the tests returns 1 at lag 4 then it cannot be determined whether the data is random.

Table 2.2 presents the results. The Ljung-Box $\mathcal{Q}$-statistic lack-of-fit hypothesis test indicates that traffic data is random ($H = 0$). Consequently, equation (2.1) is a reasonable fit to model traffic. Table 2.3 presents volatility and growth rate estimates for different wireless capacity usage time series. Bandwidth results can be found in [31].

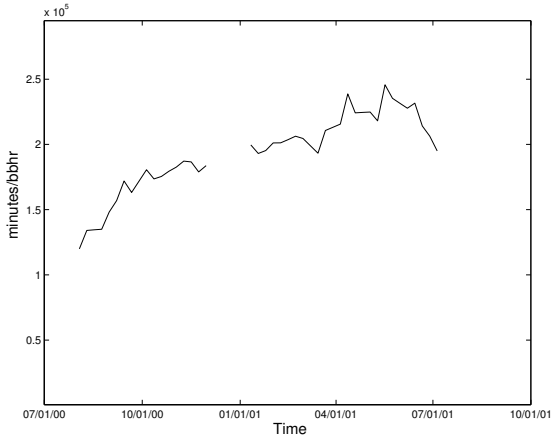
Table 2.3 Result summary table for different time series (different switches in Toronto). The Ljung-Box test is executed at the 5% significance level. The Ljung-Box test $H = 0$ indicates random data, i.e. equation (2.1) is a good fit to the data.

Time series	Drift term μ $\%/(year)$	Volatility (filtered data) σ $\%/(year)^{\frac{1}{2}}$	Ljung-Box test
A	-24.75	90.69	0
B	40.84	74.7	0
C	72.46	32.74	0

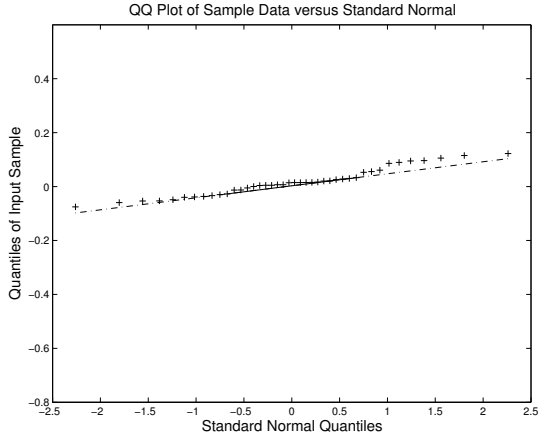
2.2.1 Jumps in Capacity Usage

Under the stochastic process described by equation (2.1), it has been possible to study the problem of optimal network investment for both bandwidth and wireless [31, 33]. While interesting results were obtained with such a simple model, the assumption of ignoring sudden large variations in capacity usage may limit our model. In particular, it is conceivable that rare events of large magnitude may affect network operators' upgrade decisions.

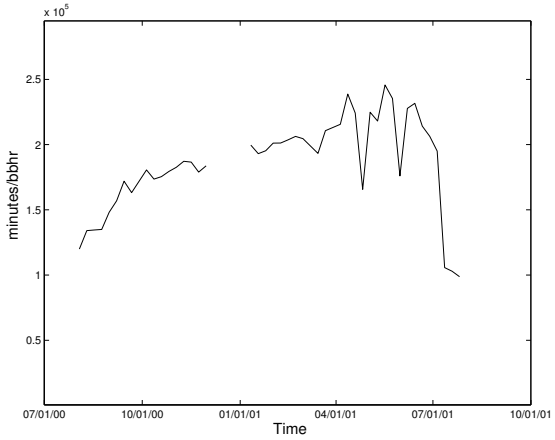
In order to determine how these rare events affect the previous model (2.1), a comparison of the capacity usage data with and without the sudden large variations is presented in Figure 2.5. It must be emphasized that in this comparison the vacations effects are still suppressed using interpolation and the month of December is ignored; the only difference are the large deviations in capacity usage which were previously suppressed.



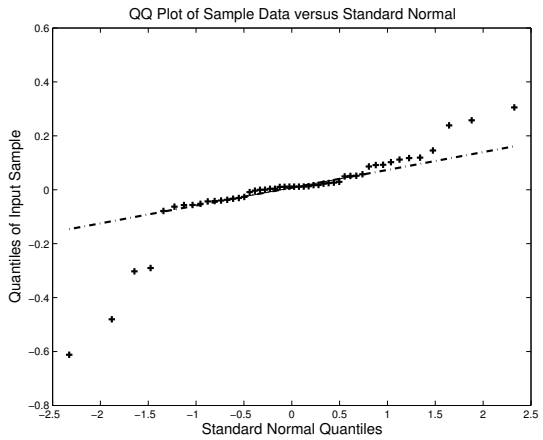
(A) *Vacation effects and suspicious changes have been suppressed using interpolation. The month of December is ignored.*



(B) *Quantile-Quantile plot of the logarithmic relative traffic changes for Figure 2.5(a). The trend of the time series has been removed.*



(C) *Vacation effects have been suppressed and the month of December is ignored. The large deviations are included.*



(D) *Quantile-Quantile plot of logarithmic relative traffic changes for Figure 2.5(c). The trend of the time series has been removed.*

FIGURE 2.5: *Quantile-Quantile plots comparing the logarithmic relative traffic changes when the jumps in capacity usage are ignored or included.*

In Figure 2.5, two Quantile-Quantile plots are presented [89]. A Quantile-Quantile plot is a common way of visualizing the difference between two distributions. The better the

fit between the two distributions, the closer the line is to straight. The first Quantile-Quantile plot (see Figure 2.5(b)) corresponds to the case when the sudden large variations in capacity usage are ignored. The line is reasonably straight, confirming once again that equation (2.1) is a good fit for capacity usage (assuming that it is valid to remove the rare events). However, when the sudden large variations in capacity usage are reintegrated into the time series, the lognormal assumption is no longer adequate. In particular it is observed that the tails of the distribution lie away from the straight line (see Figure 2.5(d)).

To alleviate this problem, equation (2.1) is augmented so as to include jumps in capacity usage [66]. In contrast with financial markets where if stocks jump in value, they tend to stay at this level, we observe that when capacity usage exhibits a sudden large variation, usage always returns very quickly to its previous level (see Figure 2.5(c)). This behaviour is not uncommon for nonfinancial instruments. The same pattern is observed for electricity spot prices [71]. But this is where the similarity between capacity usage and electricity ends. Indeed, while electricity spot prices tend to stay at around the same value over time, telecom traffic on the other hand continues to increase randomly over time. Consequently, we suggest the following process for capacity usage

$$\begin{aligned} dQ &= \alpha(\eta - Q)dt + (J-1)QdY \\ d\eta &= \eta\mu dt + \eta\sigma dZ, \end{aligned} \tag{2.4}$$

where,

Q is traffic (unit of traffic), η is the mean (unit of traffic),

μ is the growth rate (per year), σ is the volatility (year$^{-1/2}$),

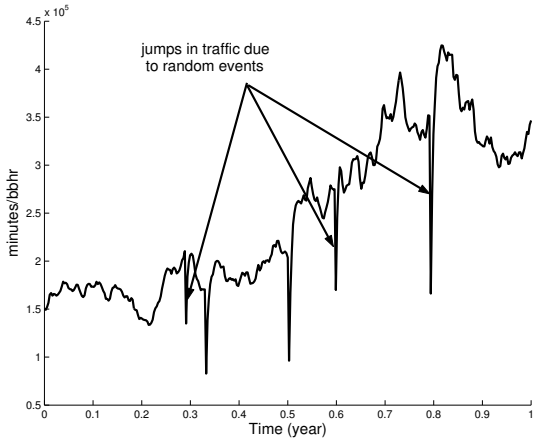
dZ is an increment of a Gauss-Wiener process, dY is a Poisson process,

$$dY = \begin{cases} 0 & \text{with probability } 1 - \lambda dt, \\ 1 & \text{with probability } \lambda dt, \end{cases}$$

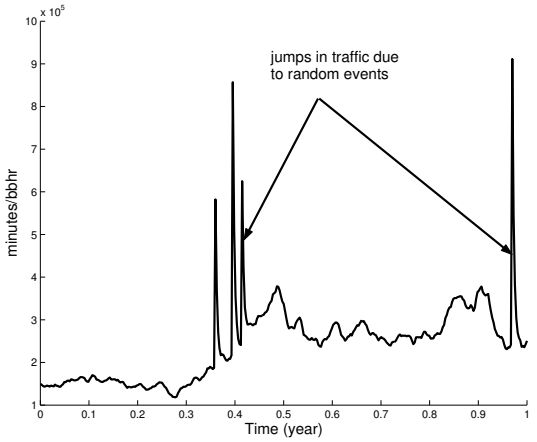
λ is the Poisson arrival intensity (per year),

α is the reversion speed (per year), and $J - 1$ produces a jump from Q to QJ

Intuitively, (2.4) implies that traffic is increasing randomly following a geometric Brownian motion stochastic process, and from time to time there are sudden temporary jumps (e.g. snow storms, power failures, earthquakes), followed by a return to the geometric Brownian motion trend.



(A) *Sudden large variations in traffic are considered to be negative on average.*



(B) *Sudden large variations in traffic are considered to be positive on average.*

FIGURE 2.6: *Illustrative sample paths followed by capacity usage over time.*

A few illustrative sample paths for capacity usage are simulated using equation (2.4).

Figure 2.6 presents the results for two different scenarios. In one case the jump sizes are on average negative (see Figure 2.6(a)), while in the other case the jump sizes are on average positive (see Figure 2.6(b)). The jump size distribution is assumed to follow a lognormal distribution as in [66]. We observe that in the case of negative jumps (see Figure 2.6 (A)), capacity usage appears to be similar to the wireless time series presented in Figure 2.6 (B).

2.3 Investment Valuation Model

Let $V(\eta, Q, t)$ be the value of an investment dependent only on Q, η and time t. Using the extended version of Itô's lemma for the jump diffusion process (2.4) [89, 35, 5], the process followed by $V(\eta, Q, t)$ is given by

$$dV = (\mathcal{G}V + R)dt + \theta dZ + \omega V dY, \tag{2.5}$$

where R represents the revenue in dollars per year,

$$\mathcal{G}V \equiv V_t + \alpha(\eta - Q)V_Q + \eta\mu V_\eta + \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta}, \tag{2.6}$$

and

$$\theta = \sigma\eta V_\eta, \quad \omega V = V(\eta, QJ, t) - V(\eta, Q, t). \tag{2.7}$$

Let us pick two investments V_1 and V_2 expiring at some future time ($> t$). From equation (2.5) we have

$$dV_1 = (\mathcal{G}V_1 + R_1)dt + \theta_1 dZ + \omega V_1 dY,$$
$$dV_2 = (\mathcal{G}V_2 + R_2)dt + \theta_2 dZ + \omega V_2 dY.$$

Both V_1 and V_2 have the same factors of uncertainty dZ and dY. Let x_1 be the fraction of the amount invested in V_1 and x_2 be the fraction of the amount invested in V_2 such that $x_1 + x_2 = 1$. The return $d\Pi$ on the portfolio Π is given by

$$\begin{aligned} d\Pi &= x_1 dV_1 + x_2 dV_2 \\ &= (x_1 \mathcal{G} V_1 + x_1 R_1 + x_2 \mathcal{G} V_2 + x_2 R_2)\, dt \\ &+ (x_1\theta_1 + x_2\theta_2)\, dZ + (x_1 \omega V_1 + x_2 \omega V_2)\, dY \end{aligned} \tag{2.8}$$

Choosing $x_1 = \theta_2$ and $x_2 = -\theta_1$, we have $x_1\theta_1 + x_2\theta_2 = 0$, then equation (2.8) becomes

$$d\Pi = (x_1 \mathcal{G} V_1 + x_1 R_1 + x_2 \mathcal{G} V_2 + x_2 R_2)\, dt + (x_1 \omega V_1 + x_2 \omega V_2)\, dY. \tag{2.9}$$

In (2.9), one of the uncertain factors dZ has been eliminated, however the other uncertain factor dY remains. In [66], the argument is made that if $\lambda \neq 0$ then it is not possible to build a riskless portfolio Π. If we are willing to assume that jumps in capacity usage can be diversifiable (i.e. not correlated with the market portfolio) then equation (2.9) can be written as

$$E[d\Pi] = (x_1 \mathcal{G} V_1 + x_1 R_1 + x_2 \mathcal{G} V_2 + x_2 R_2)\, dt + (x_1 E[\omega V_1] + x_2 E[\omega V_2])\, \lambda dt. \tag{2.10}$$

Here, $E[\cdot]$ is the expectation operator, defined by

$$E[x] = \int_0^\infty x g(x) dx,$$

and $g(J)$ is the probability density function of the jump size. Since jump risk is diversifiable, the expected return of the portfolio Π given by (2.10) must be equal to the risk free rate since

all the uncertain non-diversifiable factors have been removed. Equation (2.10) becomes

$$r(\theta_2 V_1 - \theta_1 V_2)dt = [x_1 \mathcal{G} V_1 + x_1 R_1 + x_2 \mathcal{G} V_2 + x_2 R_2 + (x_1 E[\omega V_1] + x_2 E[\omega V_2])\,\lambda]\,dt. \quad (2.11)$$

It follows that

$$\frac{\mathcal{G}V_1 + R_1 + \lambda E[\omega V_1] - rV_1}{\theta_1} = \frac{\mathcal{G}V_2 + R_2 + \lambda E[\omega V_2] - rV_2}{\theta_2}. \quad (2.12)$$

Define ζ as the value of each side of equation (2.12), i.e.

$$\frac{\mathcal{G}V_1 + R_1 + \lambda E[\omega V_1] - rV_1}{\theta_1} = \frac{\mathcal{G}V_2 + R_2 + \lambda E[\omega V_2] - rV_2}{\theta_2} = \zeta. \quad (2.13)$$

In real options theory ζ is often referred to as the market price of risk. It is a measure of the extra reward expected for taking on risk scaled by a measure of risk. Dropping the subscripts in equation (2.13), then if V is an investment dependent on Q, η and t, such that

$$dV = (\mathcal{G}V + R)dt + \theta V dZ + \omega V dY,$$

then

$$\mathcal{G}V - rV + R + \lambda E[\omega V] = \theta\zeta. \quad (2.14)$$

Substituting $\mathcal{G}$ from equation (2.6), θ and ω from equation (2.7) into equation (2.14), we find

$$V_t + \alpha(\eta - Q)V_Q + \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta} + \eta(\mu - \zeta\sigma)V_\eta - rV + R + \lambda E[V(\eta, QJ, t) - V(\eta, Q, t)] = 0. \quad (2.15)$$

Equation (2.15) can then be rewritten as

$$V_t + \alpha(\eta - Q)V_Q + R + \lambda \int_0^\infty V(\eta, QJ, t)g(J)dJ + \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta} + \eta(\mu - \zeta\sigma)V_\eta - (r+\lambda)V = 0. \quad (2.16)$$

For clarity of presentation, let us redefine $\mathcal{G}V \equiv \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta} + \eta(\mu - \zeta\sigma)V_\eta - (r+\lambda)V$. Equation (2.16) becomes

$$-V_t - \alpha(\eta - Q)V_Q = R + \lambda \int_0^\infty V(\eta, QJ, t)g(J)dJ + \mathcal{G}V. \quad (2.17)$$

Let $\tau = T - t$ (i.e. τ represents the time going backward) equation (2.17) is written as

$$V_\tau + \alpha(Q - \eta)V_Q = R + \lambda \int_0^\infty V(\eta, QJ, \tau)g(J)dJ + \mathcal{G}V. \quad (2.18)$$

Equation (2.18) is a parabolic partial integro-differential equation (PIDE).

Remark 2.1 (Market price of risk versus discount rate). *As discussed in [85], the discount rate used should reflect the riskiness of the investment. However, the riskiness of the investment depends on future decisions, and hence cannot be constant. Trigeorgis [85] argues that the market price of risk approach is the correct method to apply when the riskiness of the investment changes overtimes compared to the expected discounted cash flow method [35].*

2.4 Modelling and Numerical Issues

The goal of this thesis is to use equation (2.18) to solve the problem of optimal network investment. However, several issues remain to be addressed. From a modelling point of view, algorithms must be developed to solve the optimal network investment problem taking into

account real world constraints such as maintenance costs, upgrade costs or equipment lead time.

From a numerical point of view, equation (2.18) presents several challenges. First, equation (2.18) is a two dimensional partial integro-differential equation (PIDE) with no diffusion in the capacity usage (Q) direction. Second, equation (2.18) contains an integral term that must be handled efficiently.

In the following chapters, numerical algorithms for handling the integral term and the convection dominance problems will be presented. We develop methods for discretizing the PIDE (2.18) containing the jump integral term in a fully implicit fashion. This requires careful design of efficient methods for computing the jump integral term. A semi-Lagrangian method is used to alleviate the problem of convection dominance (no diffusion) in the capacity usage direction. Finally, we put together all these techniques to solve equation (2.18) in the context of optimal telecommunications investment.

2.5 Summary

In this chapter, we have shown that the main underlying risk factor when investing in telecommunication is capacity usage. Capacity usage is very volatile while the spot price paid by customers per unit of traffic has very low volatility. Based on these observations and some empirical studies, a two factor stochastic process was presented to model capacity usage. Using a real options approach a partial integro-differential equation was then derived for the investment value.

Chapter 3

Jump Diffusion: Algorithms

The two dimensional partial integro-differential equation (PIDE) that was previously derived in Chapter 2 presents several numerical challenges. In particular, the jump integral term must be solved accurately and quickly. In this chapter, a method to solve the jump integral term in an implicit fashion is proposed. To validate our approach, a one dimensional PIDE taken from the financial derivative pricing literature is considered.

3.1 Introduction

Before proceeding any further, let us introduce the notion of financial derivatives. Financial derivatives are contracts whose payoff depends on the outcome of future events. For example, the holder of a call option has the right but not the obligation to buy a financial asset at a fixed price at some future time. Similarly, a put option confers the right to sell a financial asset for a fixed price at some future time. Suppose that a financial asset X is currently worth \$100. An investor who owns this financial asset may purchase a put option to sell the asset at \$100 at some future time to protect his investment from a possible decline in price. However, what is the fair market value of such an option? By fair value, we

mean the value of the contract such that it cannot be bought from one party and sold to another guaranteeing a risk free profit. This valuation approach is often referred to as the no-arbitrage approach [52, 89].

Based on a no-arbitrage approach, Black and Scholes derived a linear parabolic partial differential equation (PDE) for the fair price of an option on a stock [17, 65]. However, there is increasing empirical evidence that the assumption that stocks follow a geometric Brownian motion stochastic processed is flawed [42]. To alleviate this problem numerous avenues of research have been explored which either extend the Black-Scholes model or explore completely new approaches. Among these extensive works, the jump diffusion model [66] and the stochastic volatility model (which could include jumps as well) [13, 83, 10] appear to be the most popular among practitioners.

Unfortunately, most of the existing methods for pricing options under jump processes are confined to vanilla European options [52]. There has been very little work of practical significance on numerical methods for pricing exotic or path-dependent options (e.g. discretely observed barrier, lookback, and Asian options) [89]. The method suggested by [4] is an explicit type approach based on multinomial trees. As is well-known, such methods have timestep limitations due to stability considerations, and are generally only first order correct. Zhang [92] develops a method which treats the jump integral term in explicit fashion, and the remaining terms in the PIDE implicitly. Unfortunately, rather restrictive stability conditions are required. Meyer [67] uses the method of lines to value American options where the underlying asset can jump to a finite number of states. More recently, a method based on use of a wavelet transform has been suggested by [63]. The basic idea is to use a wavelet transform to approximate the dense matrix discrete integral operator by dropping small terms. Andersen and Andreasen [5] use an operator splitting type approach combined with a fast Fourier transform (FFT) evaluation of a convolution integral to price European options with jump diffusion, where the diffusion terms involve non-constant local

volatilities. This paper [5] demonstrates that a jump diffusion model combined with a local volatility surface calibrated to real data, results in stable fitted parameters. However, it is unclear how an operator splitting approach can implicitly handle American options.

Consequently, the objective of this chapter is to present a robust and efficient numerical method for solving the jump integral term which arises from the jump diffusion model PIDE. We limit ourselves to pricing options under the jump diffusion model, but this framework is also applicable to credit risk models or more complex valuation models such as stochastic volatility with jumps.

3.2 The Basic Model

Let S represent the underlying stock price. Movements in this variable over time are assumed to be described by a jump diffusion process of the form

$$\frac{dS}{S} = \mu dt + \sigma dZ + (J-1)dY, \tag{3.1}$$

where μ is the drift rate, σ is the volatility associated with the continuous (Brownian) component of the process, dZ is the increment of a Gauss-Wiener process, dY is a Poisson process which is assumed to be independent of the Brownian part (note that $dY = 0$ with probability $1 - \lambda dt$ and $dY = 1$ with probability λdt, where λ is the Poisson arrival intensity), and $J - 1$ is an impulse function producing a jump from S to SJ.

Under equation (3.1), the stock price S has two sources of uncertainty. The term σdZ corresponds to normal levels of uncertainty while the term dY describes exceptional events. If the Poisson event does not occur ($dY = 0$), then equation (3.1) is equivalent to the usual stochastic process of geometric Brownian motion assumed in the Black-Scholes model [17].

If, on the other hand, the Poisson event occurs, then equation (3.1) can be written as

$$\frac{dS}{S} \simeq (J-1), \tag{3.2}$$

where $J-1$ is an impulse function producing a jump from S to SJ. Consequently, the resulting sample path for the stock S will be continuous most of the time with finite negative or positive jumps with various amplitudes occurring at discrete points in time.

Let $V(S,t)$ be the value of a contingent claim that depends on the underlying stock price S and time t. As is well-known, the following backward PIDE may be solved to determine V:

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r-\lambda\kappa)SV_S - rV + \left(\lambda \int_0^\infty V(SJ)g(J)dJ - \lambda V\right), \tag{3.3}$$

where $\tau = T-t$ is the time until expiry at date T, r is the continuously compounded risk free interest rate, $\kappa = E[J-1]$ denotes the expected relative jump size, and $g(J)$ is the probability density function of the jump amplitude J with the obvious properties that $\forall J$, $g(J) \geq 0$ and $\int_0^\infty g(J)dJ = 1$. For brevity, the details of the derivation of equation (3.3) have been omitted (see [66, 89, 5]) details can be found in Appendix A. An important special case is where the jump size probability distribution is lognormal [66, 84]. In that case, the probability density function is given by

$$g(J) = \frac{\exp\left(-\frac{(\log(J)-\nu)^2}{2\gamma^2}\right)}{\sqrt{2\pi}\gamma J}, \tag{3.4}$$

where ν is the mean and γ^2 is the variance of the jump size probability distribution. For future convenience, note that equation (3.3) can be rewritten in a slightly different form as

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r-\lambda\kappa)SV_S - (r+\lambda)V + \lambda \int_0^\infty V(SJ)g(J)dJ. \tag{3.5}$$

Note that, as discussed in [5], if the parameters λ, ν are obtained by calibration to market data they should be regarded as risk adjusted (i.e. $\mathcal{Q}$ measure not $\mathcal{P}$ measure).

3.2.1 Boundary Conditions

As $S \to 0$, equation (3.5) reduces to

$$V_\tau = -rV \ . \tag{3.6}$$

As $S \to \infty$, we make the common assumption that

$$V_{SS} \simeq 0 \ \ ; \ \ S \to \infty \tag{3.7}$$

which means that

$$V \simeq A(\tau)S + B(\tau) \ \ ; \ \ S \to \infty. \tag{3.8}$$

Assuming equation (3.8) holds, then equation (3.5) reduces to

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV \ \ ; \ \ S \to \infty. \tag{3.9}$$

Consequently, at both $S = 0, S \to \infty$, the PIDE (3.5) reduces to the Black-Scholes PDE, and the usual boundary conditions can be imposed. Note that in equation (3.9) V_{ss} is retained in the PDE for numerical stability purposes.

3.3 Implicit Discretization Methods

This section explores discretization methods for the PIDE, where the terms not involving the jump integral are handled implicitly. A straightforward approach to the numerical

solution of equation (3.5) would be to use standard numerical discretization methods for the non-integral terms [84], in combination with numerical integration methods such as Simpson's rule or Gaussian quadrature. However, such an approach is computationally expensive, as noted by [84]. It is more efficient to transform the integral in equation (3.5) into a correlation integral. This allows efficient FFT methods to be used to evaluate the integral for all values of S.

Let

$$\mathcal{I}(S,\tau) = \int_0^\infty V(SJ,\tau)g(J)dJ. \tag{3.10}$$

Using the change of variable

$$y = \log(SJ),\ J = \frac{\exp(y)}{S}, \text{and } dJ = \frac{\exp(y)}{S}dy, \tag{3.11}$$

and substituting (3.11) into (3.10), we obtain

$$\mathcal{I}(S,\tau) = \int_{-\infty}^\infty V\left(\exp(y),\tau\right) g\left(\frac{\exp(y)}{S}\right)\frac{\exp(y)}{S}dy. \tag{3.12}$$

Letting $f(u) = g(u)u$, equation (3.12) becomes

$$\mathcal{I}(S,\tau) = \int_{-\infty}^\infty V\left(\exp(y),\tau\right) f\left(\frac{\exp(y)}{S}\right)dy. \tag{3.13}$$

Let $x = \log(S)$, $\overline{V}(x,\tau) = V\left(\exp(x),\tau\right) = V(S,\tau)$ and $\overline{f}(\log y) = f(y)$. To avoid notational clutter, the dependence of V on τ is suppressed from now on. Substituting into equation

(3.13), we have

$$\begin{aligned}\mathcal{I} &= \int_{-\infty}^{\infty} \overline{V}(y)\overline{f}(y - log(S))dy \\ &= \int_{-\infty}^{\infty} \overline{V}(y)\overline{f}(y - x)dy \\ &= \int_{-\infty}^{\infty} \overline{V}(x + y)\overline{f}(y))dy. \end{aligned} \tag{3.14}$$

Note that $\overline{f}(x)$ is the probability density of a jump of size $x = \log J$. Equation (3.14) corresponds to the correlation product $\overline{\otimes}$ of $\overline{V}(y)$ and $\overline{f}(y)$ [77]. As a result, a discrete form of equation (3.14) can be computed efficiently using a fast Fourier transform (FFT). Equation (3.14) can be written as

$$\mathcal{I} = \overline{V}\,\overline{\otimes}\,\overline{f}. \tag{3.15}$$

If $\overline{f}$ is an even function ($\overline{f}(x) = \overline{f}(-x)$), then (3.15) corresponds to the convolution product [77].

As a specific example, consider the probability density function suggested by [66, 84]:

$$g(J) = \frac{\exp\left(-\frac{(\log(J)-\nu)^2}{2\gamma^2}\right)}{\sqrt{2\pi}\gamma J}. \tag{3.16}$$

It follows that if $f(J) = g(J)J$, then

$$\overline{f}(x) = g\left(\exp(x)\right)\exp(x) \tag{3.17}$$

$$= \frac{\exp\left(-\frac{(x-\nu)^2}{2\gamma^2}\right)}{\sqrt{2\pi}\gamma\exp(x)}\exp(x) \tag{3.18}$$

$$= \frac{\exp\left(-\frac{(x-\nu)^2}{2\gamma^2}\right)}{\sqrt{2\pi}\gamma}. \tag{3.19}$$

Let $E[\cdot]$ be the expectation operator. Then, $E[J] = \exp(\nu + \gamma^2/2)$, which means that

$\kappa = E[J-1] = \exp(\nu + \gamma^2/2) - 1$.

We can write the correlation integral (3.14) in discrete form as

$$\mathcal{I}_i = \sum_{j=-N/2+1}^{j=N/2} \overline{V}_{i+j} \overline{f}_j \Delta y + O\left((\Delta y)^2\right), \tag{3.20}$$

where $I_i = I(i\Delta x)$, $\overline{V}_j = \overline{V}(j\Delta x)$,

$$\overline{f}_j = \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \overline{f}(x) dx, \tag{3.21}$$

and $x_j = j\Delta x$. Note that we have assumed that $\Delta y = \Delta x$, and that in (3.20) N is selected sufficiently large so that the solution in areas of interest is unaffected by the application of an asymptotic boundary condition for large values of S. In particular, we assume that $\overline{V}_{N/2+j}, j > 0$ can be approximated by an asymptotic boundary condition. In practice, since $\overline{f}_j$ decays rapidly for $|j| > 0$, this should not cause any difficulty. Also note that $\overline{V}_{-N/2+j}, j < 0$, can be interpolated from known values V_k since these points represent values near $S = 0$. An important property to note is that

$$\begin{aligned} \overline{f}_j &\geq 0 \ , \quad \forall j \\ \sum_{j=-N/2+1}^{j=N/2} \overline{f}_j \Delta y &\leq 1. \end{aligned} \tag{3.22}$$

This follows because $\overline{f}(y)$ is a probability density function and $\overline{f}_j$ is defined by equation (3.21).

The discrete form of the correlation integral (3.20) uses an equally spaced grid in $\log S$ coordinates. While this is convenient for a FFT evaluation of the correlation integral, it is not particularly suitable for discretizing the PDE (details of the FFT evaluation of equation (3.20) will be given in section 3.7). We use an unequally spaced grid in S coordinates for

the PDE discretization $[S_0, \ldots, S_p]$. Let

$$V_i^n = V(S_i, \tau_n). \tag{3.23}$$

Now, $\overline{V}_j$ will not necessarily coincide with any of the discrete values V_k in equation (3.23). Consequently, we linearly interpolate (using Lagrange basis functions defined on the S grid) to determine the appropriate values, i.e. if

$$S_{\Upsilon(j)} \leq e^{j\Delta x} \leq S_{\Upsilon(j)+1}, \tag{3.24}$$

then

$$\overline{V}_j = \psi_{\Upsilon(j)} V_{\Upsilon(j)} + \left(1 - \psi_{\Upsilon(j)}\right) V_{\Upsilon(j)+1} + O\left((\Delta S_{\Upsilon(j)+1/2})^2\right), \tag{3.25}$$

where $\psi_{\Upsilon(j)}$ is an interpolation weight, and $\Delta S_{i+1/2} = S_{i+1} - S_i$. We are now faced with the problem that the integral I_i is evaluated at a point $S = e^{x_i}$ which does not coincide with a grid point S_k. We simply linearly interpolate the $\mathcal{I}_i$ to get the desired value. If $e^{x_{\Pi(k)}} \leq S_k \leq e^{x_{\Pi(k)+1}}$, then

$$\mathcal{I}(S_k) = \phi_{\Pi(k)} \mathcal{I}_{\Pi(k)} + (1 - \phi_{\Pi(k)}) \mathcal{I}_{\Pi(k)+1} + O\left((e^{x_{\Pi(k)}} - e^{x_{\Pi(k)+1}})^2\right), \tag{3.26}$$

where $\phi_{\Pi(k)}$ is an interpolation weight. Note that

$$\begin{aligned} 0 \leq \phi_i \leq 1 \\ 0 \leq \psi_i \leq 1. \end{aligned} \tag{3.27}$$

Combining equations (3.20), (3.25), and (3.26) gives

$$\mathcal{I}(S_k) = \sum_{j=-N/2+1}^{j=N/2} \chi(V,k,j)\overline{f}_j \Delta y, \tag{3.28}$$

where $V = [V_0, V_1, \ldots, V_p]'$ and

$$\begin{aligned}\chi(V,k,j) = {} & \phi_{\Pi(k)} \left[\psi_{\Upsilon(\Pi(k)+j)} V_{\Upsilon(\Pi(k)+j)} + (1-\psi_{\Upsilon(\Pi(k)+j)}) V_{\Upsilon(\Pi(k)+j)+1}\right] \\ & + (1-\phi_{\Pi(k)}) \left[\psi_{\Upsilon(\Pi(k)+1+j)} V_{\Upsilon(\Pi(k)+1+j)} + (1-\psi_{\Upsilon(\Pi(k)+1+j)}) V_{\Upsilon(\Pi(k)+1+j)+1}\right].\end{aligned} \tag{3.29}$$

For future reference, note that $\chi(V,k,j)$ is linear in V, and that if $\iota = [1,1,\ldots,1]'$, then it follows from properties (3.27) that

$$\chi(\iota,k,j) = 1 \quad \forall k,j. \tag{3.30}$$

We can now consider the complete discretization of equation (3.5). The integral term is approximated using equation (3.28). We use a fully implicit method for the usual PDE, and then use a weighted timestepping method for the jump integral term. Letting V_i^n denote the solution at node i and time level n, the discrete equations can be written as

$$\begin{aligned}& V_i^{n+1}\left[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau\right] - \Delta\tau\beta_i V_{i+1}^{n+1} - \Delta\tau\alpha_i V_{i-1}^{n+1} \\ & \quad = V_i^n + (1-\theta_J)\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(V^{n+1},i,j)\overline{f}_j\Delta y + \theta_J\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(V^n,i,j)\overline{f}_j\Delta y.\end{aligned} \tag{3.31}$$

Note that $\theta_J = 0$ corresponds to an implicit handling of the jump integral, whereas $\theta_J = 1$ indicates an explicit treatment of this term.

Discretizing the first derivative term of equation (3.5) with central differences leads to

$$\begin{aligned}\alpha_{i,central} &= \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_{i-1}} \\ \beta_{i,central} &= \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_{i-1}}.\end{aligned} \tag{3.32}$$

If $\alpha_{i,central}$ or $\beta_{i,central}$ is negative, oscillations may appear in the numerical solution. These can be avoided by using forward or backward differences at the problem nodes, leading to (forward difference)

$$\begin{aligned}\alpha_{i,forward} &= \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} \\ \beta_{i,forward} &= \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})} + \frac{(r - \lambda\kappa)S_i}{S_{i+1} - S_i},\end{aligned} \tag{3.33}$$

or (backward difference)

$$\begin{aligned}\alpha_{i,backward} &= \frac{\sigma_i^2 S_i^2}{(S_i - S_{i-1})(S_{i+1} - S_{i-1})} - \frac{(r - \lambda\kappa)S_i}{S_i - S_{i-1}} \\ \beta_{i,backward} &= \frac{\sigma_i^2 S_i^2}{(S_{i+1} - S_i)(S_{i+1} - S_{i-1})}.\end{aligned} \tag{3.34}$$

Algorithmically, we decide between a central, forward discretization at each node for equation according to Algorithm 3.1.

Note that the test condition in Algorithm 3.1 guarantees that α_i and β_i are non-negative. For typical parameter values and grid spacing, forward or backward differencing is rarely required for single factor options. In practice, since this occurs at only a small number of nodes remote from the region of interest, the limited use of a low order scheme does not result in poor convergence as the mesh is refined. As we shall see, requiring that all α_i and β_i are non-negative has important theoretical ramifications.

As $S \to 0$, equation (3.3) reduces to $V_\tau = -rV$, which is simply incorporated into the

Algorithm 3.1 For each node i, central, forward or backward discretization is selected such that α_i and β_i are positive

if $\alpha_{i,central} \geq 0$ and $\beta_{i,central} \geq 0$ **then**
  $\alpha_i = \alpha_{i,central}$
  $\beta_i = \beta_{i,central}$
else if $\beta_{i,forward} \geq 0$ **then**
  $\alpha_i = \alpha_{i,forward}$
  $\beta_i = \beta_{i,forward}$
else
  $\alpha_i = \alpha_{i,backward}$
  $\beta_i = \beta_{i,backward}$
end if

discrete equations (3.31) by setting $\alpha_i = \beta_i = 0$ at $S_i = 0$. In practice we truncate the S grid at some large value $S_p = S_{\max}$, where we impose Dirichlet conditions at $p = imax$. These Dirichlet conditions can be determined by substituting equation (3.8) into equation (3.9), and using the option payoff as an initial condition. This is done by replacing equation (3.31) at $S = S_{\max} = S_p$ with the specification that V_p^{n+1} is equal to the relevant Dirichlet condition.

We now proceed to consider the stability of the discretization (3.31). In particular, the following result is obtained:

Theorem 3.1 (Stability of scheme (3.31)). *The discretization method* (3.31) *is unconditionally stable for any choice of* $\theta_J, 0 \leq \theta_J \leq 1$*, provided that*

- $\alpha_i, \beta_i \geq 0$*;*
- *the discrete probability density* $\overline{f}_j$ *has the properties* (3.22)*;*
- *the interpolation weights satisfy* (3.27)*;*
- $r, \lambda \geq 0$*;*
- *a Dirichlet boundary condition is imposed at* $S = S_{\max}$*.*

Proof. Let $V^n = [V_0^n, V_1^n, \ldots, V_p^n]'$ be the discrete solution vector to equation (3.31). Suppose the initial solution vector is perturbed, i.e.

$$\hat{V}^0 = V^0 + E^0, \tag{3.35}$$

where $E^n = [E_0^n, \ldots, E_p^n]'$ is the perturbation vector. Note that $E_p^n = 0$ since Dirichlet boundary conditions are imposed at this node. Then we obtain the following equation for the propagation of the perturbation (noting that χ is a linear operator)

$$\begin{aligned} &E_i^{n+1}\left[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau\right] - \Delta\tau\beta_i E_{i+1}^{n+1} - \Delta\tau\alpha_i E_{i-1}^{n+1} \\ &\quad = E_i^n + (1-\theta_J)\Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(E^{n+1}, i, j)\overline{f}_j \Delta y + \theta_J \Delta\tau\lambda \sum_{j=-N/2+1}^{j=N/2} \chi(E^n, i, j)\overline{f}_j \Delta y, \\ &\qquad\qquad i = 0, \ldots, p-1. \end{aligned} \tag{3.36}$$

Defining

$$\|E\|_\infty^n = \max_i |E_i|^n, \tag{3.37}$$

it follows from properties (3.22), (3.27), and (3.30) and $\alpha_i, \beta_i \geq 0$ that

$$\begin{aligned} |E_i^{n+1}|\left[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau\right] &\leq \|E\|_\infty^n + (1-\theta_J)\Delta\tau\lambda\|E\|_\infty^{n+1} + \theta_J\Delta\tau\lambda\|E\|_\infty^n \\ &\quad + \Delta\tau\beta_i|E_{i+1}^{n+1}| + \Delta\tau\alpha_i|E_{i-1}^{n+1}|, \end{aligned} \tag{3.38}$$

$$i = 0, \ldots, p-1. \tag{3.39}$$

This implies

$$\begin{aligned} |E_i^{n+1}|\left[1 + (\alpha_i + \beta_i + r + \lambda)\Delta\tau\right] &\leq (\Delta\tau\beta_i + \Delta\tau\alpha_i)\|E\|_\infty^{n+1} \\ &\quad + \|E\|_\infty^n + (1-\theta_J)\Delta\tau\lambda\|E\|_\infty^{n+1} + \theta_J\Delta\tau\lambda\|E\|_\infty^n. \end{aligned} \tag{3.40}$$

Now, equation (3.40) is valid for all $i < p$. In particular, it is true for node m, where

$$\max_i |E_i^{n+1}| = |E_m^{n+1}|. \tag{3.41}$$

Writing equation (3.40) for $i = m$ gives

$$\|E\|_\infty^{n+1} \left[1 + (r + \theta_J \lambda)\Delta\tau\right] = \|E\|_\infty^n (1 + \theta_J \Delta\tau\lambda), \tag{3.42}$$

and thus

$$\begin{aligned} \|E\|_\infty^{n+1} &\leq \|E\|_\infty^n \frac{(1 + \theta_J \Delta\tau\lambda)}{(1 + (r + \theta_J\lambda)\Delta\tau)} \\ &\leq \|E\|_\infty^n. \quad \square \end{aligned} \tag{3.43}$$

Remark 3.1 (Unconditional stability with explicit evaluation of the integral). *When discretizing the correlation integral term explicitly ($\theta_J = 1$) the scheme presented in equation* (3.31) *remains unconditionally stable. Note that in [92] a conditionally stable method was developed. The conditional stability was a result of a slightly different timestepping approach compared to that in equation* (3.31).

3.4 Crank-Nicolson Discretization

The discretization method used in the previous section is only first order correct in the time direction. In order to improve the timestepping error, we can use a Crank-Nicolson method.

Such an approach results in the following set of discrete equations

$$V_i^{n+1}\left[1+(\alpha_i+\beta_i+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta_i V_{i+1}^{n+1}-\frac{\Delta\tau}{2}\alpha_i V_{i-1}^{n+1}$$
$$=V_i^n\left[1-(\alpha_i+\beta_i+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta_i V_{i+1}^n+\frac{\Delta\tau}{2}\alpha_i V_{i-1}^n$$
$$+(1-\theta_J)\lambda\Delta\tau\sum_{j=-N/2+1}^{j=N/2}\chi(V^{n+1},i,j)\overline{f}_j\Delta y$$
$$+\theta_J\lambda\Delta\tau\sum_{j=-N/2+1}^{j=N/2}\chi(V^n,i,j)\overline{f}_j\Delta y. \quad (3.44)$$

A full Crank-Nicolson method is obtained by setting $\theta_J = 1/2$ in equation (3.44). If we define the matrix M such that

$$-[MV^n]_i = V_i^n(\alpha_i+\beta_i+r+\lambda)\frac{\Delta\tau}{2}-\frac{\Delta\tau}{2}\beta_i V_{i+1}^n-\frac{\Delta\tau}{2}\alpha_i V_{i-1}^n-\frac{\Delta\tau}{2}\lambda\sum_{j=-N/2+1}^{j=N/2}\chi(V^n,i,j)\overline{f}_j\Delta y, \quad (3.45)$$

then we can write equation (3.44) as

$$[I-M]\,V^{n+1}=[I+M]\,V^n. \quad (3.46)$$

Alternatively, we can define $D=[I-M]^{-1}[I+M]$, so that equation (3.46) can be written as

$$V^n=(D)^nV^0. \quad (3.47)$$

Note that the notation $(D)^n$ refers to D raised to the power of n. Consequently, an initial perturbation vector E^0 will generate a perturbation at the nth step, E^n, given by $E^n = (D)^nE^0$.

The stability of the operator D is defined in terms of the power boundedness of D. If n is the number of timesteps and p is the number of grid nodes, then given some matrix

norm $\|\cdot\|$, we say that D is *strictly stable* if

$$\|(D)^n\|_\infty \leq 1 \quad \forall n, p. \tag{3.48}$$

Following [48], *strong stability* is defined as

$$\|(D)^n\|_\infty \leq C \quad \forall n, p, \tag{3.49}$$

and *algebraic stability* is defined as

$$\|(D)^n\|_\infty \leq C n^s p^q \quad \forall n, p. \tag{3.50}$$

where $C, s, q \geq 0$ are constants independent of n and p.

Algebraic stability is obviously a weaker condition than either strict or strong stability. Note that the Lax Equivalence Theorem states that strong stability is a necessary and sufficient condition for convergence for all initial data. Weaker algebraic stability yields convergence only for certain initial data. For a more detailed discussion of this, see [48].

If μ_i are the eigenvalues of D, then a necessary condition for strong stability is that $|\mu_i| \leq 1$, and that any $|\mu_i| = 1$ has multiplicity one. From equation (3.45) and properties (3.22), we have that

- The off-diagonals of M are all non-negative.
- The diagonals of M (excluding the last row) are strictly negative.
- Assuming that $r > 0$, $\sum_{j=0}^{j=p} M_{ij} < 0$ for $i = 0, \ldots, p-1$.
- The last row of M is identically zero due to the Dirichlet boundary condition.

It then follows that all the Gerschgorin disks of M are strictly contained in the left half of the complex plane, with one eigenvalue identically zero. Hence all the eigenvalues of D are

strictly less than one in magnitude, with one eigenvalue having modulus one. As a result, D satisfies the necessary conditions for strict stability.

However, since D is non-symmetric, this is not sufficient for power boundedness of D [18]. As discussed in [56] and [58], we can guarantee algebraic stability by examining the γ numerical range of the matrix M. In the case $\gamma = 1$, the numerical range of M coincides with the convex hull of the Gerschgorin disks of M when the maximum norm is used in equation (3.50). These results can be summarized in the following theorem:

Theorem 3.2 (Algebraic Stability of Crank-Nicolson Timestepping). *If the discretization satisfies the conditions of Theorem 3.1, then the Crank-Nicolson discretization* (3.44) *is algebraically stable in the sense that*

$$\|(D)^n\|_\infty \leq Cn^{1/2} \quad \forall n, p,$$

where C is independent of n, p.

Proof. Since all the Gerschgorin disks of M are in the left half of the complex plane, this follows from the results in [58]. □

In fact, we believe that the algebraic stability estimate is overly pessimistic. For the case of constant coefficients with a log-spaced grid, in Appendix C we show using Von Neumann analysis that Crank-Nicolson timestepping with the correlation product is unconditionally strictly stable. However, it is interesting to note that if we use Crank-Nicolson weighting for the PDE terms and an explicit method for the jump diffusion term ($\theta_J = 1$ in equation (3.44)), then a Von Neumann analysis shows that this method is only conditionally stable ($\lambda\Delta\tau$ must be sufficiently small).

3.5 Fixed Point Iteration Method

When using an implicit discretization, it is computationally inefficient to solve the full linear system because the correlation product term makes the system dense. Consequently, in this section we explore the use of a fixed point iteration scheme to solve the linear system which results from an implicit discretization of the correlation product term. This idea was suggested in [84], but no convergence analysis was given.

Define the matrix L such that

$$-[LV^n]_i = V_i^n(\alpha_i + \beta_i + r + \lambda)\Delta\tau - \Delta\tau\beta_i V_{i+1}^n - \Delta\tau\alpha_i V_{i-1}^n, \tag{3.51}$$

and the matrix B such that

$$[B \cdot V^n]_i = \sum_i b_{ij} V_j^n = \sum_{j=-N/2+1}^{j=N/2} \chi(V^n, i, j)\overline{f}_j \Delta y. \tag{3.52}$$

Note that $[B \cdot V^n]$ is a linear function of V^n. Thus we can write a fully implicit ($\theta = 0$) or Crank Nicolson ($\theta = 1/2$) discretization as

$$[I - (1-\theta)L]V^{n+1} = [I + \theta L]V^n + (1-\theta)\lambda\Delta\tau BV^{n+1} + \theta\lambda\Delta\tau BV^n. \tag{3.53}$$

Note that the entries of the matrix B have the property

$$0 \le b_{ij} \le 1 \text{ and } \sum_j b_{ij} \le 1, \tag{3.54}$$

a fact that will be important in the convergence analysis later. Dirichlet boundary conditions

are enforced at $i = p$ by setting

$$\begin{aligned} L_{ij} &= 0 \;\; ; \;\; i = p \\ b_{ij} &= 0 \;\; ; \;\; i = p \\ V_p^{n+1} &= V_p^n. \end{aligned} \tag{3.55}$$

In order to avoid algebraic complication, we assume that the Dirichlet condition at S_p is independent of time. We can then derive the fixed point iteration method presented in Algorithm 3.2. Note that in Algorithm 3.2, the matrix vector multiply $B\hat{V}^k$ is computed in $N\log(N)$ flops using a fast Fourier transforms (FFT).

Algorithm 3.2 Fixed point iteration scheme to evaluate the jump integral term implicitly.

Let $(V^{n+1})^0 = V^n$
Let $\hat{V}^k = (V^{n+1})^k$
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Solve
  $[I - (1-\theta)L]\,\hat{V}^{k+1} = [I + \theta L]\,V^n + (1-\theta)\lambda\Delta\tau B\hat{V}^k + \theta\lambda\Delta\tau BV^n$
  if $\max_i \frac{|\hat{V}_i^{k+1} - \hat{V}_i^k|}{\max(1, |\hat{V}_i^{k+1}|)} < tolerance$ **then**
    Quit
  end if
end for

Letting $e^k = V^{n+1} - \hat{V}^k$, the convergence of the fixed point scheme (see Algorithm 3.2) can be summarized in the following theorem:

Theorem 3.3 (Convergence of the fixed point iteration). *Provided that*

- $\alpha_i, \beta_i \geq 0$ *(see Section 3.3);*
- *the discrete probability density* $\overline{f}_j$ *has the properties* (3.22)*;*
- *the interpolation weights satisfy* (3.27)*;*
- $r \geq 0, \lambda \geq 0$*;*

- *Dirichlet boundary conditions* (3.55) *are used*

then the fixed point iteration Algorithm 3.2 is globally convergent, and the maximum error at each iteration satisfies

$$\|e^{k+1}\|_\infty \leq \|e^k\|_\infty \frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(r+\lambda)\Delta\tau}.$$

Proof. It is easily seen from the iteration Algorithm 3.2 that e^k satisfies

$$[I-(1-\theta)L]\, e^{k+1} = (1-\theta)\lambda\Delta\tau B e^k. \tag{3.56}$$

Following the same steps used to prove Theorem 3.1, we therefore obtain

$$\begin{aligned}\|e^{k+1}\|_\infty &\leq \|e^k\|_\infty \frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(r+\lambda)\Delta\tau} \\ &< \|e^k\|_\infty. \quad \square\end{aligned} \tag{3.57}$$

Note that typically $\lambda\Delta\tau \ll 1$, so that

$$\|e^{k+1}\|_\infty \simeq \|e^k\|_\infty (1-\theta)\lambda\Delta\tau, \tag{3.58}$$

which will result in rapid convergence. It is also interesting to observe that the number of iterations required for convergence is independent of the number of nodes in the S grid.

3.6 American Options

In the following, we extend our numerical framework to handle American options. An American option has the additional feature that it is permitted to be exercised at any time during its lifetime. The valuation of American options is therefore more challenging since

at each timestep, given S, the value of the option must be found and it must be determined whether or not it is optimal to exercise. This problem is often referred to as the *free boundary problem* [89].

Previous work on numerical methods for American options under jump diffusion used an explicit timestepping method for the jump integral term, and a standard linear complementarity solver to solve the algebraic linear complementarity problem at each timestep [92]. This method is only first order correct in time, and conditionally stable. In contrast, the method in [5] is second order correct in time for Bermudan (discretely early exercisable) options.

We develop an implicit timestepping approach, which has the potential of second order accuracy in the time direction. We use a penalty method [44, 93] to enforce the American constraint. As discussed in [44], a penalty method can be easily extended to multifactor models [93], and to nonlinear models such as uncertain volatility and transaction costs [73]. It is a common misconception that penalty methods result in poorly conditioned algebraic problems. This is shown not to be the case in [44].

If we define

$$\mathcal{L}V \equiv V_\tau - \left(\frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - (r + \lambda) V + \lambda \int_0^\infty V(SJ) g(J) dJ \right) \tag{3.59}$$

and if $V^*(S, \tau)$ is the payoff, then the American option pricing problem can be stated as

$$\begin{aligned} \mathcal{L}V &\geq 0 \\ (V - V^*) &\geq 0 \\ (\mathcal{L}V = 0) \vee (V - V^* &= 0) \end{aligned} \tag{3.60}$$

where the notation $(\mathcal{L}V = 0) \vee (V - V^* = 0)$ denotes that either $(\mathcal{L}V = 0)$ or $(V - V^* = 0)$

at each point in the solution domain. Equations (3.60) can be written more compactly as [70]

$$\min(\mathcal{L}V; V - V^*) \quad = \quad 0 \; . \tag{3.61}$$

Remark 3.2 (Viscosity solution). *It is well known that there is in general no smooth solution to equation (3.61). In what follows it will be understood that we are seeking viscosity solutions to equation (3.61). We note that the degeneracy of $\mathcal{L}V$ as $S \to 0$ causes no particular difficulty in the case of viscosity solutions. A detailed discussion of the existence and uniqueness of viscosity solutions to equation (3.61), can be found in [70, 3]. In addition, sufficient conditions to ensure convergence of a discrete numerical scheme to the viscosity solution, is given in [11]. Finally, an application of the results in [11] to the case of European options with jump diffusion is given in [21]. Due to the non-local property of the integral term, care must be taken when considering viscosity solutions. A localization procedure is used in [24] in order to precisely characterize viscosity solutions for jump models.*

3.6.1 The Penalty Method

The basic idea of the penalty method is simple. Equation (3.5) is replaced by the nonlinear PIDE [41]

$$V_\tau = \frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - (r + \lambda) V + \lambda \int_0^\infty V(SJ) g(J) dJ + \rho \max(V^* - V, 0), \tag{3.62}$$

where, in the limit as the positive penalty parameter $\rho \to \infty$, the solution satisfies $V \geq V^*$.

As shown in [44], in the case where $\lambda = 0$ (no jumps) the penalty method can be used to obtain an approximate solution to the discretized complementarity problem (3.60) at each timestep. For details regarding the penalty method, we refer the reader to [44].

Let V^* be the vector of payoffs obtained upon exercise, and let the diagonal matrix P

be given by

$$P(V^{n+1})_{ii} = \begin{cases} Large & \text{if } V_i^{n+1} < V_i^* \\ 0 & \text{otherwise.} \end{cases} \tag{3.63}$$

Then the matrix form of the discrete equations for the penalized method is given by

$$[I - (1-\theta)L + P(V^{n+1})]V^{n+1} = \\ [I + \theta L]V^n + (1-\theta_J)\lambda\Delta\tau BV^{n+1} + \theta_J\lambda\Delta\tau BV^n + \left[P(V^{n+1})\right]V^*. \tag{3.64}$$

Dirichlet boundary conditions are enforced at $i = p$ by setting

$$\begin{aligned} L_{ij} &= 0 \;\; ; \;\; i = p \\ P_{ij} &= 0 \;\; ; \;\; i = p \\ b_{ij} &= 0 \;\; ; \;\; i = p \\ V_p^{n+1} &= V_p^n. \end{aligned} \tag{3.65}$$

In order to avoid algebraic complication, we assume that the Dirichlet condition at S_p is independent of time.

Remark 3.3 (Stability of a fully implicit discretization). *It is straightforward to show, via a maximum analysis, that setting $\theta = 0$ in equation* (3.64) *results in an unconditionally stable method for any θ_J, $0 \leq \theta_J \leq 1$.*

3.6.2 The Matrix Iteration

In order to solve equation (3.64), Algorithm 3.2 is augmented as to include the penalty vector P (assuming $\theta = \theta_J$).

The matrix vector multiplies in Algorithm 3.3 ($B\hat{V}^k$) can be computed in $O(N \log N)$

Algorithm 3.3 Fixed point iteration scheme to solve the free boundary problem under the diffusion process.

Let $(V^{n+1})^0 = V^n$
Let $\hat{V}^k = (V^{n+1})^k$
Let $\hat{P}^k = P((V^{n+1})^k)$
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Solve

$$\left[I - (1-\theta)L + \hat{P}^k\right] \hat{V}^{k+1} = [I + \theta L] V^n + \hat{P}^k V^* + (1-\theta)\lambda\Delta\tau B\hat{V}^k + \theta\lambda\Delta\tau B V^n$$

  if $\max_i \frac{|\hat{V}_i^{k+1} - \hat{V}_i^k|}{\max(1, |\hat{V}_i^{k+1}|)} <$ tolerance **then**
    Quit
  end if
end for

operations using an FFT (see Section 3.7). Consequently, the work for each iteration is dominated by the forward and backward FFTs.

3.6.3 Convergence of the Iteration

In the following, we consider the problem of convergence of the iteration scheme (3.3). For clarity, the proofs of the different lemmas and theorems can be found in Appendix D. Our main result can be summarized in the following theorem

Theorem 3.4 (Convergence of iteration (3.3)). *Let L, B and $\hat{P}^k$ be given by* (3.51), (3.52) *and* (3.63), *respectively. Assume that $\alpha_i \geq 0, \beta_i \geq 0$ in equation* (3.51), *that B has the properties* (3.54) *and that we use a Dirichlet boundary condition in* (3.65). *Then the iteration Algorithm 3.3 is globally convergent to the unique solution of equation* (3.64) *for any initial iterate $\hat{V}^0$.*

Proof. See Appendix D. □

3.7 Details Regarding Evaluation of the Correlation Integral

To complete the discussion of our numerical algorithm, we need to consider issues such as evaluating the jump integral term, interpolation, and wrap-around effects. Note that each iteration of the Algorithm 3.2 requires evaluation of a correlation integral for all points on the PDE grid.

Fast evaluation of this integral using FFT methods necessitates transformation to an equally spaced grid in $x = \log(S)$ coordinates. If the original PDE grid is equally spaced in $\log(S)$, then there is clearly no difficulty. However, this type of grid spacing is highly inefficient for cases involving discontinuous payoffs or barriers. We therefore prefer not to restrict the type of grid used for the original PDE. Recall that the correlation integral is

$$\mathcal{I}(x) = \int_{-\infty}^{\infty} \overline{V}(x+y)\overline{f}(y)dy,$$

or, in discrete form

$$\mathcal{I}_i = \sum_{j=-N/2+1}^{j=N/2} \overline{V}_{i+j}\overline{f}_j\Delta y + O\left((\Delta y)^2\right),$$

where $\mathcal{I}_i = \mathcal{I}(i\Delta x)$, $\overline{V}_j = \overline{V}(j\Delta x)$, and $\overline{f}_j$ is defined by equation (3.21). We have also assumed that $\Delta y = \Delta x$, and that $\overline{V}(\log S) = V(S)$.

Now, $\overline{V}_j$ will not necessarily coincide with any of the discrete values V_k in equation (3.31). Consequently, we will linearly interpolate to determine the appropriate values, as in equation (3.25). Since equation (3.20) has the form of a discrete correlation, FFT methods are an obvious choice to compute this efficiently. Assuming that $\overline{f}$ is real, then

$$\mathrm{FFT}(\mathcal{I})_k = (\mathrm{FFT}(\overline{V}))_k(\mathrm{FFT}(\overline{f}))^*_k, \tag{3.66}$$

where $(\cdot)^*$ denotes the complex conjugate. Since $\overline{f}(z)$ is the probability density of $z = \log J$,

which is a specified function, we can simply precompute FFT($\overline{f}$) on the required equally spaced grid in z coordinates. We can then carry out an inverse FFT to obtain the values of the correlation integral on the equally spaced $x = \log S$ grid. A further interpolation step is required to obtain the value of the correlation integral on the original S grid (equation (3.26)).

We can summarize the steps needed to generate the required values $\mathcal{I}(S_k)$, $k = 0, \ldots, p$ into Algorithm 3.4.

Algorithm 3.4 Summary of the steps needed to generate the required values $I(S_k)$, $k = 0, \ldots, p$ using a fast Fourier transform.

1 Interpolate the discrete values of V onto an equally spaced $\log(S)$ grid . This generates the required values of $\overline{V}_j$.
2 Carry out the FFT on this data.
3 Compute the correlation in the frequency domain (with precomputed $FFT\left(\overline{f}\right)$), using equation (3.66).
4 Invert the FFT of the correlation.
5 Interpolate the discrete values of $\mathcal{I}(x_i)$ onto the original S grid.

Note that as long as linear or higher order interpolation is used, this procedure is second order correct, which is consistent with the discretization error in the PDE and the midpoint rule used to evaluate the integral (3.20).

In principle, we can avoid the interpolation steps in the above procedure if we use special techniques for computing the FFT for unequally spaced data. There are several methods for computing the inverse FFT problem (i.e. given unequally spaced data, determine the Fourier coefficients), as well as the forward FFT problem (given the Fourier coefficients, determine the inverse transform values on an unequally spaced grid) [88, 37, 76]. However, it should be noted that we are not particularly interested in obtaining highly accurate estimates of the discrete Fourier coefficients, as we simply need to evaluate the correlation integral correct to second order. A discussion of methods for applying FFT techniques to unequally spaced data is given in Appendix F. An alternative approach, based on a Fast

Gauss transform is discussed in Appendix E. For our purposes there is no particular benefit in terms of accuracy in using these other methods. We will use the interpolation method (3.4) followed by the standard FFT to calculate our illustrative results in Chapter 4.

Another issue requiring attention is that the FFT algorithm effectively assumes that the input functions are periodic. This may cause wrap-around pollution unless special care is taken when implementing the algorithm. The integral (3.13) is approximated on the finite domain

$$\mathcal{I}(x) = \int_{y_{\min}}^{y_{\max}} \overline{V}(x+y)\overline{f}(y)dy. \tag{3.67}$$

The PDE part of the PIDE (3.5) is computed using the finite computational domain $[0, S_{\max}]$, using the discrete grid $S_0, S_1, \ldots, S_{\max}$. Initially, we chose

$$\begin{aligned} y_{\max} &= \log(S_{\max}) \\ y_{\min} &= \log(S_1), \end{aligned} \tag{3.68}$$

assuming $S_1 > 0$. Note that $y_{\min} = \log(S_1)$ since normally $S_0 = 0$, so that $\log(S_0) = -\infty$.

Generally, $\overline{f}(y)$ (which represents the probability density of a jump of $S \to SJ$ where $y = \log J$) is rapidly decaying for $|y| \gg 0$. However, $\overline{V}(y)$ does not decay to zero near $y = y_{\min}, y_{\max}$. Typically, $V(S) \leq$ Const. S as $S \to \infty$, and $V(S) \simeq$ Const. as $S \to 0$, or in $y = \log S$ coordinates,

$$\overline{V}(y) \leq \text{Const. } e^y, \quad y \to \infty \tag{3.69}$$

$$\simeq \text{Const.}, \quad y \to -\infty. \tag{3.70}$$

This will cause undesirable wrap-around effects if we use an FFT approach to evaluate the integral (3.67), since the discrete Fourier transform (DFT) is effectively applied to the periodic extension of the input functions. To avoid these problems, we extend the domain

of the integral to the left and right by a size which reflects the width of the probability density. In other words, we use the values $\overline{V}(y), y \in [y_{\min} - \Delta y^-, y_{\max} + \Delta y^+]$ as input to the correlation evaluation (3.66).

In order to determine values in the extended region, we solve the following PDE-PIDE in the region $[0, S_{max}e^{\Delta y^+}]$.

$$
\begin{aligned}
V_\tau &= -rV \\
&\qquad S = 0 \\
V_\tau &= \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r+\lambda)V + \lambda \int_0^\infty V(S\eta)g(\eta)d\eta \qquad (3.71) \\
&\qquad 0 < S < S_{max} \\
V_\tau &= \frac{1}{2}\sigma^2 S^2 V_{SS} + rSV_S - rV \qquad (3.72) \\
&\qquad S_{max} \le S \le S_{max}e^{\Delta y^+} \quad .
\end{aligned}
$$

The validity of equation (3.72) can be justified by assuming that $V(S)$ is a linear function of S in the extended domain. This is a common assumption in financial application. The extended region $[S_{max}, S_{max}e^{\Delta y^+}]$ can be regarded as a buffer zone which reduces the effect of FFT wrap-around. Note that we have assumed that S_{max} is sufficiently large so that it is valid to assume that approximation (3.9) holds.

The values of $\overline{V}(u)$ for $u \in [y_{\max}, y_{\max} + \Delta y^+]$ are estimated using simple linear interpolation. The values in the left extension can be determined from interpolation on the original S grid.

This extended domain is then used as input to the forward FFT, the correlation computation (in the spectral domain), and the inverse FFT. The values in the domain extensions are affected by wrap-around and are discarded. This causes no difficulty in the right extension. In the left extension, we actually need an estimate of the value $\overline{V}(S_0 = 0)$. This is obtained from the solution of $V_\tau = -rV$ at $S = 0$. In Appendix B, we show how to estimate

$\Delta y^+, \Delta y^-$ so that the errors due to wrap-around are within a user specified tolerance.

An alternative approach for evaluating the correlation integral could involve a fast Gauss transform (FGT) [50]. This method has complexity of $O(N)$ and does not require an equally spaced grid. The use of this method has been explored in the general option pricing context by [22]. In the particular case of jump diffusion, this approach would work for the case where the jump size is lognormally distributed. It is not clear if it could be applied for other jump distributions. We have carried out some numerical experiments using public domain FGT software, and it appears that the FFT approach used here is superior to the use of FGT, at least for grid sizes of practical interest (details are presented in Appendix E). Note that all of the theoretical results given previously would be unchanged if the FGT were used instead of the FFT.

3.8 Remarks about the Viscosity Solution

Viscosity solution theory [11] allows us to show uniform convergence to the solution without requiring a priori smoothness of the solution (e.g. digital payoffs [75]) or non-degeneracy of coefficients. In particular, conditions for numerical schemes which guarantee convergence to the viscosity solution were developed by [11]. For a full description of the terminology and technical details, readers are referred to [11] and the references therein. The main result of [11] is presented in Theorem 3.5.

Theorem 3.5 ([11], page 12). *Given a PDE which satisfies the strong comparison principle, then a discretization scheme which is consistent, stable and monotone converges to the viscosity solution.*

Remark 3.4. *Although convergence is guaranteed, the results of Barles [11] says nothing about the rate of convergence. Barles [11] result was extended to the case of non-linear PIDEs in [21, 24].*

Normally any reasonable discretization scheme is consistent, although there are some technical problems in proving consistency on finite computational domains for PIDE's, due to the nonlocal nature of the jump integral term. See [21, 24] for details of proofs of consistency for trapezoidal rule evaluation of the jump integral term on a finite computational domain.

The most interesting requirement for convergence of a discrete scheme to the viscosity solution is monotonicity. A scheme which is monotone also satisfies a discrete comparison principle which means that the discrete solution satisfies certain no-arbitrage inequalities [24]. As a result, in the following the monotonicity of scheme (3.62) is investigated.

We begin by considering equation (3.62) discretized fully implicitly. For clarity of presentation, the fully implicit discretization ($\theta = 0$, (3.64)) is rewritten in matrix form as

$$[I - \Delta\tau L - \lambda\Delta\tau B]V^{n+1} = V^n + P(V^{n+1})(V^\star - V^{n+1}). \tag{3.73}$$

At each node (S_i), equation (3.73) can be written as

$$g_i(V_i^{n+1}, \{V_j^{n+1}\}_i, V_i^n) = V_i^n - [(I - \Delta\tau L - \lambda\Delta\tau B)V^{n+1}]_i + P(V_i^{n+1})(V^\star - V_i^{n+1}) = 0 \tag{3.74}$$

where $\{V_j^{n+1}\}_i$ is to be interpreted as the set of values $V_j^{n+1}, j \neq i$ and $j = 1, \ldots, p$.

Definition 3.1 (Monotone discretizations). *A discretization of the form* (3.73) *is monotone if*

$$g_i(V_i^{n+1}, \{V_j^{n+1}\}_i + \rho_j^{n+1}, V_i^n + \rho_i^n) \geq g_i(V_i^{n+1}, \{V_j^{n+1}\}_i, V_i^n) \quad \forall i; \quad \forall j \neq i$$
$$\forall \rho_j^n \geq 0, \ \forall \rho_j^{n+1} \geq 0, \tag{3.75}$$
$$g_i(V_i^{n+1} + \rho_i^{n+1}, \{V_j^{n+1}\}_i, V_i^n) < g_i(V_i^{n+1}, \{V_j^{n+1}\}_i, V_i^n) \quad \forall i; \forall j \neq i$$
$$\forall \rho_i^{n+1} \geq 0 \tag{3.76}$$

Remark 3.5. *In the viscosity solution literature [11], only condition (3.75) is used to define monotonicity. However, in the conservation law literature [59, 46] monotonicity is usually defined including condition (3.76).*

Lemma 3.1 (Monotonicity of the discretization (3.74)**).** *The fully implicit discretization (3.74) is unconditionally monotone.*

Proof. From Lemma D.3, matrix $[I-\Delta\tau L-\lambda\Delta\tau B]$ is an M matrix. As a result $-[(I-\Delta\tau L-\lambda\Delta\tau B)V_j^{n+1}]_i$ is a non-decreasing function of $\{V_j^{n+1}\}_i$. The term $P(V_i^{n+1})(V^\star - V_i^{n+1})$ is a non-increasing function of V_i^{n+1}. Similarly, $-[(I-\Delta\tau L-\lambda\Delta\tau B)V_j^{n+1}]_i$ is a decreasing function of V_i^{n+1}. Hence the discretization is monotone from Definition 3.1. □

Remark 3.6 (Extension to uncertain volatility/transaction costs). *It is simple to extend scheme* (3.74) *to the case of a nonlinear model with uncertain volatility or transaction costs. Based on Remark 3.2 and the results in [73], it is then straightforward to show convergence to the viscosity solution.*

Crank-Nicolson is only conditionally monotone, hence only conditionally satisfies the sufficient conditions in [11, 21]. However, if the Rannacher modification of Crank-Nicolson is used, then, consistent with the results in [73], our experimental computations indicate that fully implicit and Crank-Nicolson timestepping converge to the same solution. We conjecture that convergence to the viscosity solution can be obtained under somewhat weaker conditions than strict monotonicity as suggested in [11].

3.9 Summary

The main results of this chapter are as follows:

- We proved that the jump diffusion term can be discretized explicitly, and, when coupled with a fully implicit treatment of the usual PDE, the resulting timestepping

method is unconditionally stable.

- We proved that a simple fixed point iteration scheme can be used to solve the discretized algebraic equations, and that this iteration is globally convergent. In fact, for typical values of the timestep size and Poisson arrival intensity, the l_∞ error is reduced by two orders of magnitude at each iteration.

- We also developed a method for efficiently computing the jump integral term. We made no assumptions about the form of the probability density for the jump term except that the density had finite activity [66, 24]. This general approach required the evaluation of correlation type integrals, as in [92]. We also showed how to eliminate the wrap-around effects which often plague FFT methods. The correlation integral term can be rapidly computed using FFT methods.

- In contrast with previous work, we did not assume that the grid was equally spaced in either the underlying asset price or its logarithm. This was a major advantage for the pricing of contracts with barrier provisions, which typically require fine grid spacing near barriers in order to achieve sufficient accuracy.

- We developed an iterative method for solving the discrete penalized equations which result from the discretization of the differential-integral complementarity problem for pricing American options on assets which follow a jump diffusion process. We have also derived sufficient conditions for the global convergence of this iteration scheme (at each timestep).

- Unlike previous work, the method developed here uses implicit timestepping for both the correlation integral term and the American constraint. As a result, we may obtain higher order convergence (in terms of timestepping error) compared with previous methods which treat the correlation integral or the American constraint explicitly.

A major advantage of the method developed here is that it is straightforward to add a jump process to existing option pricing software. In particular, existing software, which uses an implicit approach for valuing American options, can be simply modified to price American options with jump diffusion. Non-linear pricing models (transaction costs, uncertain volatility) can also be easily extended to model jumps, as well as path dependent pricing problems (Asian, Parisian) [95].

Chapter 4

Jump Diffusion: Numerical Results

In Chapter 3, an implicit method was developed for the numerical solution of option pricing models where the underlying process follows a jump diffusion process. Proofs of timestepping stability and convergence of a fixed point iteration scheme were provided. In the following, numerical tests of convergence for a variety of options are presented.

4.1 Results

This section presents numerical results for various options, including vanilla European and American options, digital options, and options with barrier features. Unless stated otherwise, we use the Crank-Nicolson discretization scheme (3.44). The discrete system of equations is solved using the fixed point iteration Algorithm 3.2 with a convergence tolerance of tol $= 10^{-6}$. For the American option tests, the penalty term is set to 1/tol [44].

We begin by considering European options under the assumptions that the continuous part of the underlying stock price process follows geometric Brownian motion and that the proportional jump size is lognormally distributed, where the jump size distribution $g(J)$ is

given by

$$g(J) = \frac{e^{\left(-\frac{(\log(J)-\nu)^2}{2\gamma^2}\right)}}{\sqrt{2\pi}\gamma J}. \tag{4.1}$$

This allows us to check the accuracy of our algorithm against the analytic solution of [66]. Table 4.1 contains the input parameters. These are roughly the same as those estimated by [5] using European call options on the S&P 500 stock index in April of 1999.

Table 4.1 Input data used to value European options under the lognormal jump diffusion process. These parameters are approximately the same as those reported in [5] using European call options on the S&P 500 stock index in April of 1999.

σ	0.15	r	0.05
γ	0.45	ν	-0.90
λ	0.10	T	0.25
K	100.00		

We are particularly interested in the convergence properties of the algorithm as the grid is refined. For each test, as we double the number of grid points we cut the timestep size ($\Delta\tau = .01$ on the coarsest grid) in half. The convergence ratio presented in the tables below is defined in the following way. Let

$$\Delta\tau = \max_n(\tau^{n+1} - \tau^n),$$
$$\Delta S = \max_i(S_{i+1} - S_i).$$

Note that we are allowing here for the possibility of using variable timestep sizes (to be explained below), although most of our tests will simply use a constant timestep size. If we then carry out a convergence study letting $h \to 0$ where $\Delta S = \text{Const. } h$, $\Delta\tau = \text{Const. } h$, then we can assume that the error in the solution (at a given node) is $V_{\text{approx}}(h) = V_{\text{exact}} +$

Const. h^{ξ}, and the convergence ratio is defined as

$$R = \frac{V_{\text{approx}}(h/2) - V_{\text{approx}}(h)}{V_{\text{approx}}(h/4) - V_{\text{approx}}(h/2)}. \tag{4.2}$$

In the case of quadratic convergence ($\xi = 2$), then $R = 4$, while for linear convergence ($\xi = 1$), $R = 2$.

Table 4.2 Value of a European put option at $S = 100$ using Crank-Nicolson timestepping for linear, quadratic and cubic interpolation. The interpolation schemes are used to transfer data between the non-uniform S grid and the uniform log-spaced FFT grid. The input parameters are provided in Table 4.1. The convergence ratio R is defined in equation (4.2). The exact solution is 3.149026. The number of points used for the FFT grid is 2^{α}, where α is the smallest integer such that the number of nodes in the non-uniform S grid $p \leq 2^{\alpha}$.

Size of S grid	No. of Timesteps	Linear		Quadratic		Cubic	
		Value	R	Value	R	Value	R
128	25	3.146361	n.a.	3.145896	n.a.	3.146361	n.a.
255	50	3.148354	n.a.	3.148249	n.a.	3.148354	n.a.
509	100	3.148856	3.973	3.148831	4.039	3.148832	4.175
1017	200	3.148983	3.949	3.148977	3.990	3.148977	3.287
2033	400	3.149015	4.001	3.149014	4.007	3.149014	3.997
4065	800	3.149023	3.997	3.149023	4.002	3.149023	3.997

Recall that interpolation is required to transform data from the clustered PDE grid to the equally spaced $\log S$ grid, and vice versa. In Table 4.2, we compare linear interpolation (see equations (3.25)-(3.26)) with quadratic and cubic Lagrange interpolation for a vanilla European put option with different number of points on the FFT grid.

In Table 4.2 we observe quadratic convergence to the exact solution for all three interpolation schemes. Note that our earlier theoretical analysis for stability and convergence of the fixed point iteration was based on linear interpolation. This was required because linear interpolation is the only Lagrange interpolation method which has non-negative weights.

Table 4.3 Value of a European call option using Crank-Nicolson timestepping. The input parameters are provided in Table 4.1. The convergence ratio R is defined in equation (4.2). The exact solution is 0.527638 at $S = 90$, 4.391246 at $S = 100$, and 12.643406 at $S = 110$. The number of points used for the FFT grid is 2^α, where α is the smallest integer such that the number of nodes in the non-uniform S grid $p \leq 2^\alpha$. Quadratic interpolation is used.

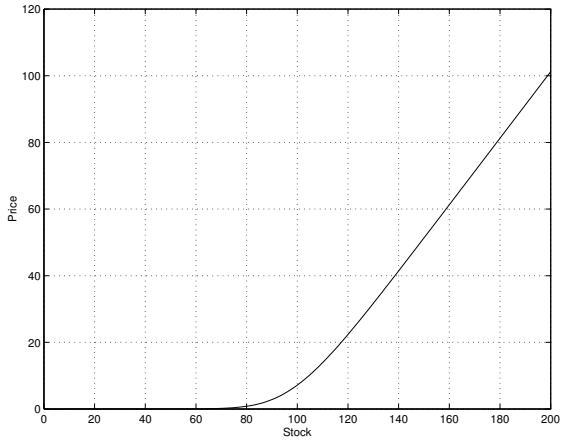
Size of	No. of	$S = 90$		$S = 100$		$S = 110$	
S grid	Timesteps	Value	R	Value	R	Value	R
128	25	0.526562	n.a.	4.388091	n.a.	12.641501	n.a.
255	50	0.527379	n.a.	4.390462	n.a.	12.642942	n.a.
509	100	0.527574	4.186	4.391050	4.039	12.643290	4.125
1017	200	0.527622	4.042	4.391197	3.991	12.643377	4.008
2033	400	0.527634	4.046	4.391233	4.005	12.643399	4.059
4065	800	0.527637	4.023	4.391243	4.002	12.643404	4.049

Although it is not the case for these particular parameter values, our numerical experiments indicate that quadratic interpolation is often more accurate than linear interpolation (although the rate of convergence is theoretically the same for both methods). Consequently, in all subsequent examples we will use quadratic interpolation. In Table 4.3 we show the convergence rate for a call option using the data in Table 4.1. For each value of S in Table 4.3, we observe second order convergence.

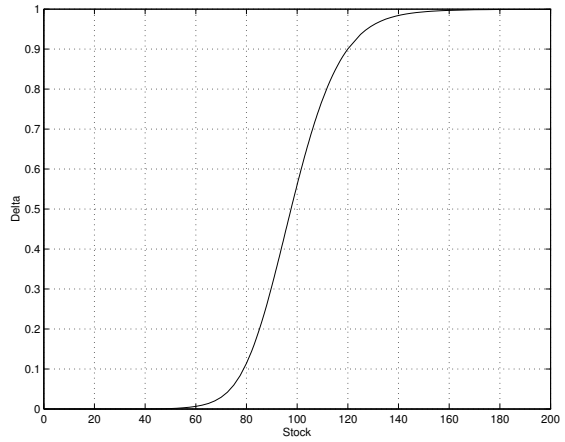
Next, we present the delta V_S and gamma V_{SS} of the solution of a European call option in Figure 4.1. In both figures, we observe that the option values, delta and gamma are smooth.

4.1.1 Non-Smooth Payoffs

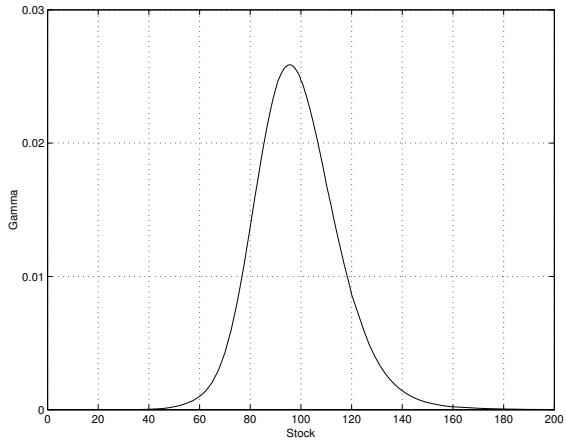
We now consider the issues raised by the presence of a discontinuity in the payoff. Oscillations are more likely to be a problem in this context if we use Crank-Nicolson timestepping, and, unless care is taken, rates of convergence can be reduced. A detailed discussion of this can be found in [75] for the case without jumps. Following [78], it is possible to restore quadratic convergence if any discontinuities in the payoff (arising either due to the



(A) *Call option price.*



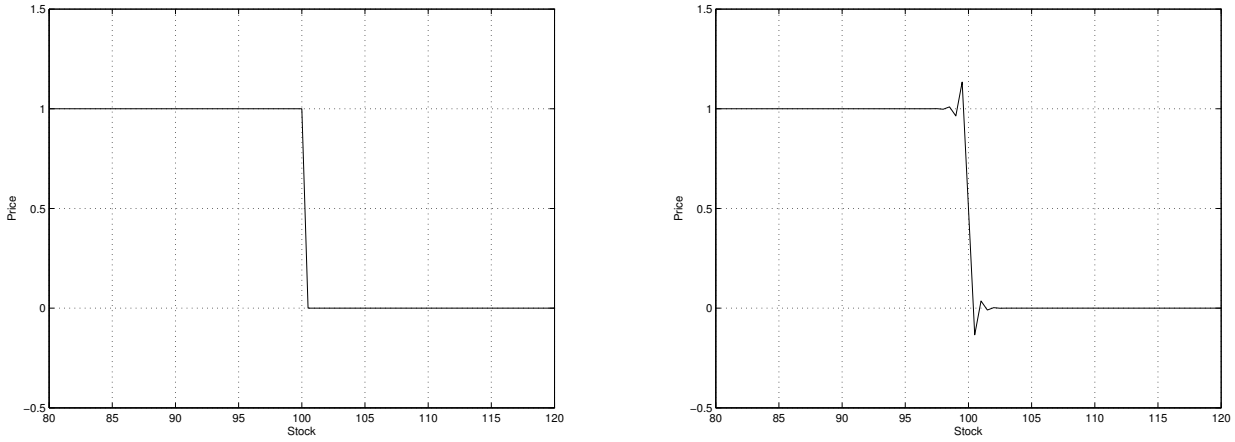
(B) *Call option delta.*



(C) *Call option gamma.*

FIGURE 4.1: *Call option values, delta (V_S) and gamma (V_{SS}) for Crank-Nicolson timestepping with* 128 *points with initial timestep* 0.01*. The input data is contained in Table 4.1.*

payoff function itself in the case of a digital option, or from the application of a discretely observed barrier) are l_2 projected onto the space of linear Lagrange basis functions, and a fully implicit method is used for a small number of timesteps after any discontinuities arise. We will refer to this technique as Rannacher timestepping. While this method does ensure



(A) *Digital call payoff.*

(B) *Digital call projected payoff.*

FIGURE 4.2: *The digital payoff figure 4.2(a) is l_2 projected onto the space of basis functions 4.2(b). This approach combined with Rannacher timestepping restores quadratic convergence [75].*

quadratic convergence, it does not guarantee the absence of oscillations. Typically, however, the use of the fully implicit timesteps smooths out the function enough that oscillations are not a problem.

Table 4.4 Value of a European digital put option using Rannacher timestepping and l_2 projection. The input parameters are provided in Table 4.1. The convergence ratio R is defined in equation (4.2). The exact solution is 0.854898 at $S = 90$, 0.387153 at $S = 100$, and 0.077923 at $S = 110$. The number of points used for the FFT grid is 2^α, where α is the smallest integer such that the number of nodes in the non-uniform S grid $p \leq 2^\alpha$. Quadratic interpolation is used.

Size of	No. of	$S = 90$		$S = 100$		$S = 110$	
S grid	Timesteps	Value	R	Value	R	Value	R
128	25	0.855540	n.a.	0.387139	n.a.	0.077539	n.a.
255	50	0.855058	n.a.	0.387151	n.a.	0.077830	n.a.
509	100	0.854938	3.988	0.387152	7.157	0.077899	4.160
1017	200	0.854908	4.059	0.387153	2.219	0.077917	3.896
2033	400	0.854900	3.990	0.387153	3.590	0.077922	3.970
4065	800	0.854899	3.991	0.387153	4.124	0.077923	4.005

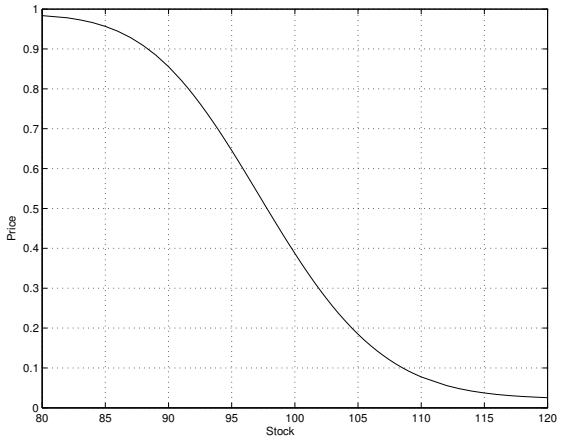
We now investigate the application of Rannacher timestepping in the jump diffusion context for a digital put option which pays \$1 at maturity if the underlying stock price is below the strike price, and zero otherwise. Table 4.4 presents a convergence study for the digital put with jumps, using Rannacher timestepping (with two fully implicit steps) and l_2 projection. As shown in this table, quadratic convergence is generally achieved, though perhaps a bit more erratically than for the vanilla payoff as shown in Table 4.3. Figure 4.3 provides plots of the solution value for a digital put along with the hedging parameters delta (V_S) and gamma (V_{SS}).

We continue our numerical tests by pricing a butterfly spread using call options (i.e. purchasing a low strike call, selling two middle strike calls, and buying a high strike call). This provides an interesting test because the payoff pattern has a sharp kink at the middle strike.

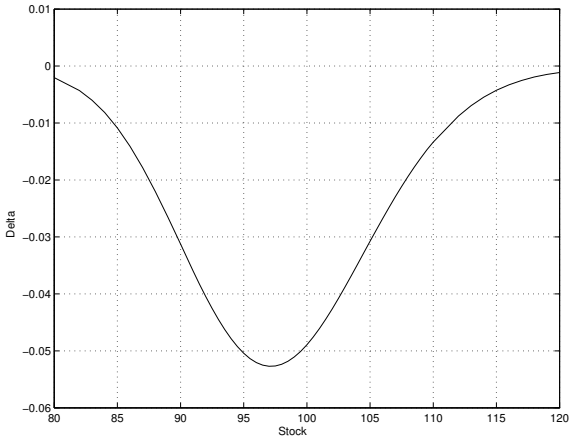
Table 4.5 Value of a European butterfly call option option using Rannacher timestepping. The input parameters are provided in Table 4.1, except that the options used have strike prices of 90, 100, and 110. The convergence ratio R is defined in equation (4.2). The exact solution is 2.280396 at $S = 95$, 2.397189 at $S = 100$, and 2.004577 at $S = 110$. The number of points used for the FFT grid is 2^α such that α is the smallest integer $p \leq 2^\alpha$, where p is the number of nodes in the non-uniform S grid. Quadratic interpolation is used.

Size of	No. of	$S = 95$		$S = 100$		$S = 110$	
S grid	Timesteps	Value	R	Value	R	Value	R
128	25	2.279206	n.a.	2.396078	n.a.	2.002380	n.a.
255	50	2.280140	n.a.	2.396983	n.a.	2.004147	n.a.
509	100	2.280352	4.396	2.397166	4.947	2.004494	5.093
1017	200	2.280385	6.565	2.397182	11.412	2.004557	5.510
2033	400	2.280394	3.509	2.397188	3.043	2.004573	3.939
4065	800	2.280396	4.437	2.397189	5.125	2.004576	4.228

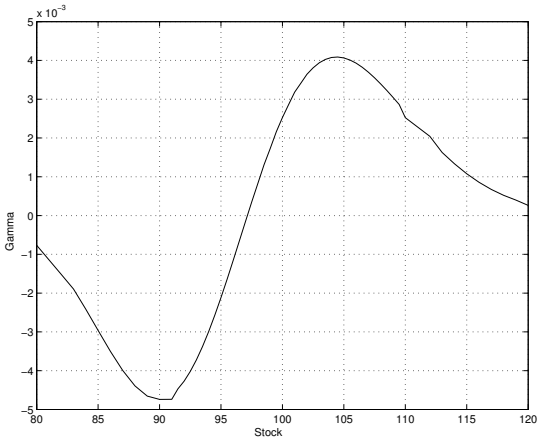
Table 4.5 presents our convergence test results. Rannacher timestepping is used. We observe quadratic convergence, though it is somewhat erratic. When comparing the numerical and analytic solution, we note that our algorithm produces very accurate answers, even



(A) Digital put value.



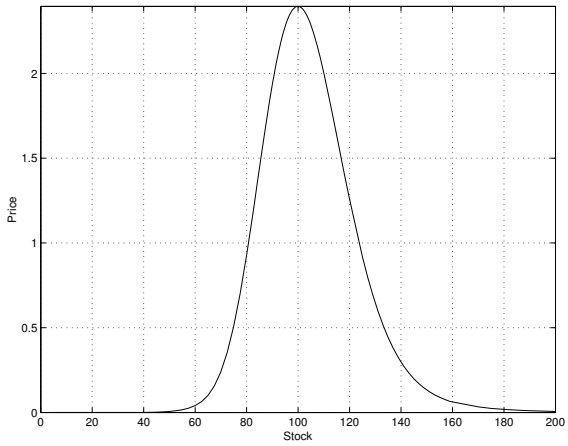
(B) Digital put delta.



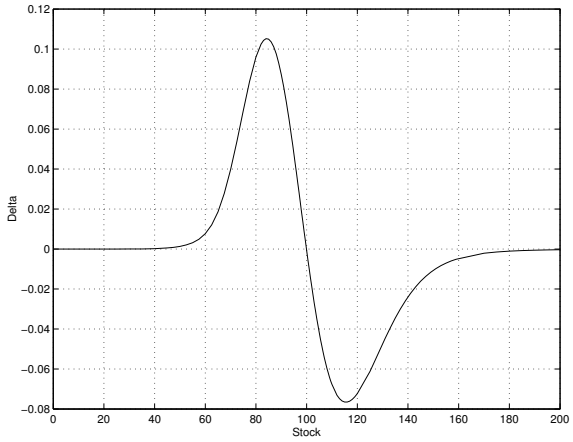
(C) Digital put gamma.

FIGURE 4.3: *Digital put option value, delta and gamma for Rannacher timestepping. The input parameters are provided in Table 4.1.*

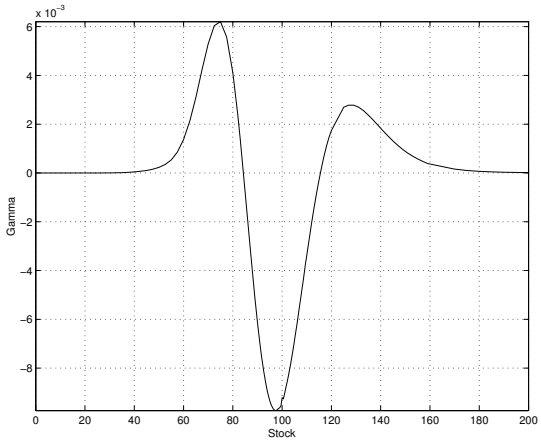
on the coarsest initial grid with 128 nodes and 25 timesteps. No oscillations are present in the solution or its derivatives (see Figure 4.4).



(A) *Butterfly call option price.*



(B) *Butterfly call option delta.*



(C) *Butterfly call option gamma.*

FIGURE 4.4: *Butterfly call option values, delta and gamma for Crank-Nicolson timestepping with* 128 *points with initial timestep* 0.01*. The input data is contained in Table 4.1. Two Rannacher timesteps are used to smooth the initial data.*

4.1.2 Probability Density function

Our next numerical tests incorporate the use of an automatic timestep size selector as described in [54]. It is not generally possible to achieve second order convergence for American

options using constant timesteps [44]. An initial timestep is given and the next timestep is computed according to

$$\Delta\tau^{n+2} = \left(\min_i \left[\frac{\text{dnorm}}{\frac{|V(S_i,\tau^n+\Delta\tau^{n+1})-V(S_i,\tau^n)|}{\max(\ \text{D},|V(S_i,\tau^n+\Delta\tau^{n+1})|,|V(S_i,\tau^n)|)}} \right] \right) \Delta\tau^{n+1}, \tag{4.3}$$

where dnorm is a target relative change (during the timestep) specified by the user. The scale D prevents the timestep selector from taking an excessive number of timesteps in regions where the value is small. In general it is set to D = 1.0 for options valued in dollars.

The first test to incorporate variable timesteps involves an alternative distribution for the jump size. Kou [55] suggests the double exponential distribution for the log jump size, observing that it has desirable analytical properties. In the model of [55],

$$\overline{f}(x) = p\eta_1 \exp(-\eta_1 x)\mathcal{H}(x) + q\eta_2 \exp(\eta_2 x)\mathcal{H}(-x), \tag{4.4}$$

where $\eta_1 > 1$, $\eta_2 > 0$, $p > 0$, $q = 1 - p > 0$, and $\mathcal{H}(\cdot)$ is the Heaviside function. As noted in [55], the condition $\eta_1 > 1$ is used to ensure that the proportional jump and stock price have finite expectation. In this model, $\kappa = E[J - 1] = \frac{p\eta_1}{\eta_1 - 1} + \frac{q\eta_2}{\eta_2 + 1} - 1$.

To provide a basis for comparison with the lognormal distribution, we attempted to find parameters for the double exponential distribution which match those used for the lognormal distribution given in Table 4.1. This did not work well for those parameters, as the mean is too far below zero, resulting in only the left tail of the double exponential being used. To remedy this, we shifted the lognormal mean from its value of $-.90$ in Table 4.1 to $-.10$. We then performed a numerical search to find parameters to match the first three central moments of the two distributions as closely as possible. We obtained values of $p = 0.3445$, $\eta_1 = 3.0465$, and $\eta_2 = 3.0775$. Figure 4.5 shows the double exponential probability density

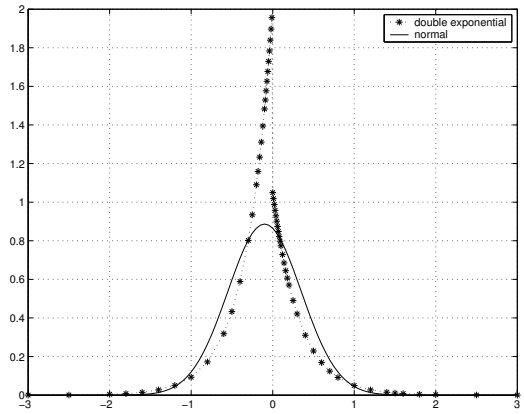
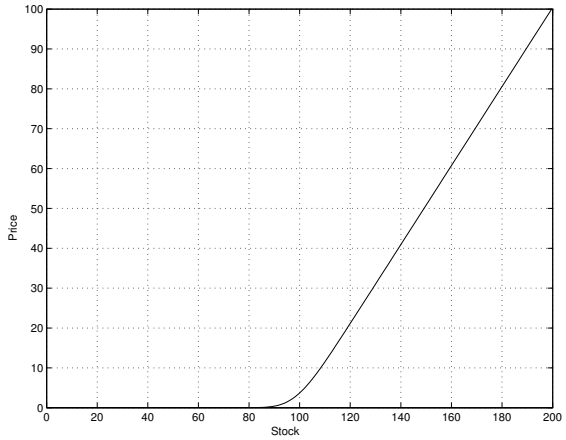


FIGURE 4.5: *Overall comparison of the normal ($\mu = -.10, \gamma = .45$) and double exponential probability density functions ($p = 0.3445$, $\eta_1 = 3.0465$, $\eta_2 = 3.0775$).*

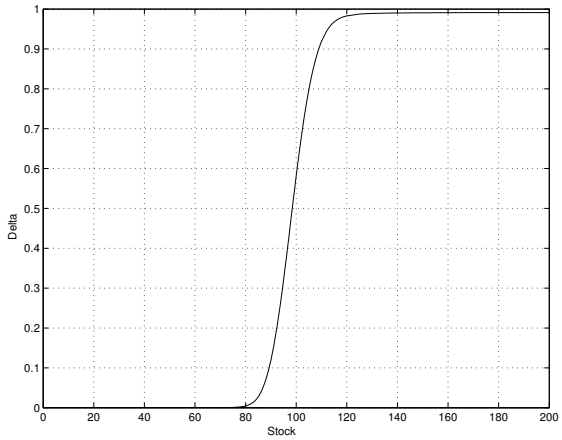
function and the normal probability density function for our parameter values. Note that the double exponential distribution has a discontinuity at zero. This can be expected to cause some problems for our numerical integration using an FFT method.

Table 4.6 Value of a European vanilla call option using Rannacher timestepping with variable timestep sizes for the double exponential probability density function (4.4). The timesteps are selected using equation (4.3), with dnorm = 0.1 on the coarsest grid, and divided by two for each grid refinement. The input parameters are $\sigma = 0.15$, $r = 0.05$, $\lambda = 0.1$, $T = 0.25$, $K = 100$, $\eta_1 = 3.0465$, $\eta_2 = 3.0775$, and $p = .3445$. The convergence ratio R is defined in equation (4.2). The exact solution is 0.672677 at $S = 90$, 3.973479 at $S = 100$, and 11.794583 at $S = 110$. The number of points used for the FFT grid is $8 \times 2^\alpha$, where α is the smallest integer such that the number of nodes in the non-uniform S grid $p \leq 2^\alpha$. Quadratic interpolation is used.

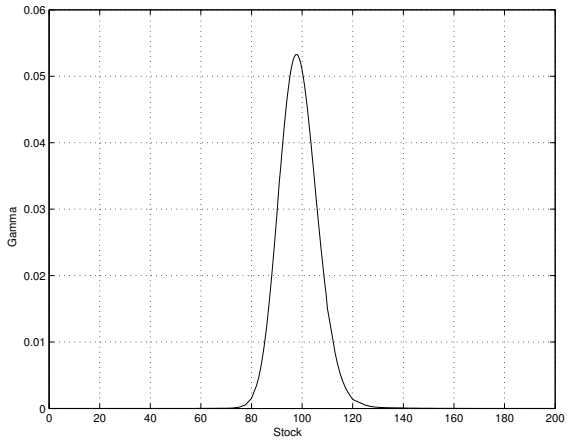
Size of	No. of	$S = 90$		$S = 100$		$S = 110$	
S grid	Timesteps	Value	R	Value	R	Value	R
128	34	0.671314	n.a.	3.969969	n.a.	11.78927	n.a.
255	65	0.672213	n.a.	3.972476	n.a	11.79248	n.a.
509	132	0.672535	2.791	3.973107	3.972	11.79367	2.688
1017	266	0.672630	3.358	3.973322	2.936	11.79416	2.431
2033	533	0.672660	3.225	3.973407	2.511	11.79438	2.244
4065	1067	0.672670	2.917	3.973445	2.281	11.79448	2.130



(A) Call option value.



(B) Call option delta.



(C) Call option gamma.

FIGURE 4.6: *Call option value, delta and gamma for Rannacher timestepping using the double exponential probability density function* (4.4)*. The input parameters are provided in the caption to Table 4.6.*

Table 4.6 presents numerical convergence tests for pricing a European call option. In an attempt to deal with the discontinuity at zero, the number of points used on the uniform-spaced x grid has been oversampled to a greater extent than in the lognormal case. In particular, the number of points on the FFT grid is $8 \times 2^{\alpha}$, where α is the smallest integer

such that 2^α is at least equal to the number of nodes in the S grid. Rannacher timestepping is used. In contrast to our earlier examples, we do not obtain second order convergence here. Instead the results indicate convergence at a linear (or perhaps slightly higher) rate to the exact solution. Despite the discontinuity, we observe smooth solution plots for the solution value, delta, and gamma in Figure 4.6.

Note that other numerical experiments indicate that we can achieve quadratic convergence in the double exponential case if we restrict the parameters so that the distribution is continuous at zero (i.e. set $p = 0.50, \eta_1 = \eta_2$). This still requires a heavily oversampled FFT grid relative to the lognormal case in order to adequately capture the sharp peak of the distribution.

4.1.3 American Numerical Examples

In this section we give a number of numerical examples which illustrate the performance and convergence of our iteration scheme for American options. The examples are chosen to demonstrate that for practical parameters values, the iterative method for solving the discrete nonlinear algebraic equations at each timestep converges rapidly. In fact, the number of iterations required for convergence of European options (with jumps) is on average, almost the same as the corresponding American option. We also verify that quadratic convergence is obtained as the grid and timesteps are refined, for Crank-Nicolson timestepping. In [44] the authors showed experimentally that in order to restore quadratic convergence when pricing American put options, a timestep selector must be used. Consequently, as in [44] we use a timestep selector (4.3) based on a modified form of that suggested in [54].

American Put Option Example

As a first example, we consider the case of an American put option under a jump diffusion process.

Table 4.7 Value of an American put, under jump diffusion process, $S = 100$, $t = 0$. The input parameters are provided in Table 4.1. The label *Itns* is the total number of iterations required in algorithm (3.3), for all timesteps. The label *Change* is the change from one level of refinement and the next. The label *Ratio* is the ratio of changes. Crank-Nicolson (modified using Rannacher [79] smoothing) and fully implicit timestepping is used with the timestep selector defined by (4.3), where dnorm = .05 and the initial timestep $\Delta\tau = .005$, on the coarsest grid. Grid and timestep sizes are reduced by half on each refinement.

Size of	No. of	Crank Nicolson			
S grid	Timesteps	Itns	Value	Change	R
127	40	121	3.2373512	n.a.	n.a.
254	100	239	3.2404239	.0030727	n.a.
508	218	507	3.2410657	.0006418	4.8
1016	453	1044	3.2412099	.0001442	4.5
2032	924	2106	3.2412435	.0000336	4.3

Size of	No. of	Fully Implicit			
S grid	Timesteps	Itns	Value	Change	R
127	40	122	3.2257718	n.a.	n.a.
254	100	254	3.2349932	.0092214	n.a.
508	218	507	3.2383450	.0033518	2.8
1016	454	1044	3.2398533	.0015083	2.2
2032	924	2060	3.2405629	.0007099	2.1

Table 4.7 shows the results for a convergence study. The timestep selector (4.3) is used. The Rannacher modification for Crank-Nicolson timestepping suggested in [79] (initial two steps fully implicit, followed by Crank-Nicolson thereafter) is used, since the payoff is non-smooth. Table 4.7 also shows the results for fully implicit timestepping.

Bates [14] developed an analytic approximation for pricing American options under the jump diffusion process of [66]. When comparing with our numerical algorithm, we found that the analytic approximation of [14] is quite accurate for the out of the money case where $S = 110$, about eight cents too low when $S = 100$, and around seventy cents too low for the in the money case with $S = 90$. This suggests that (at least for our parameter values),

Bates' [14] approximation is not very accurate (in terms of absolute pricing error), unless the option is deep out of the money.

Since the ratio of changes R in Table 4.7 appears to be approaching four as the grid and timesteps are refined, this indicates that convergence is approximately quadratic in ΔS and $\Delta\tau$. Table 4.7 also indicates that the average number of iterations per timestep for algorithm (3.3) is of the order $2-3$.

Figure 4.7 compares the jump diffusion solution (jumps) for an American option with a constant volatility Black-Scholes solution (no-jumps). To ensure a consistent basis for comparison, the volatility used in the no-jump model is the implied volatility which reproduces the jump model price at $S = K = 100$. For a European call $\sigma_{\text{imp}} = .1886$. Note that the jump model is significantly more valuable than the non-jump model at $S = 110$, due to the high probability that a downward jump in the asset price can occur. Figure 4.8 shows the delta (V_S) and gamma (V_{SS}) for the jump and no-jump models.

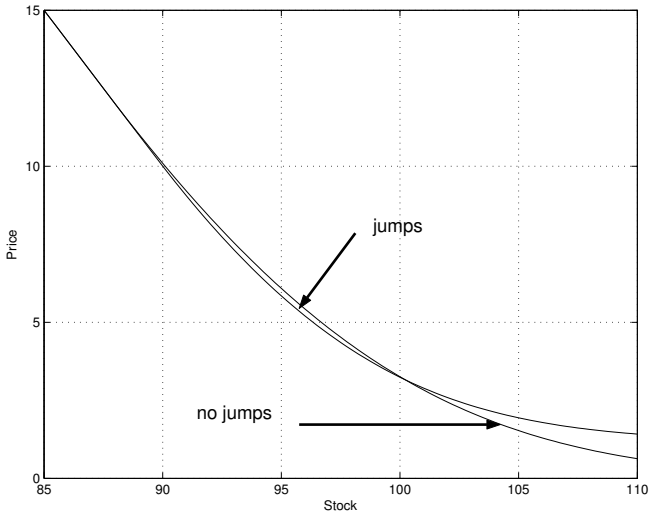
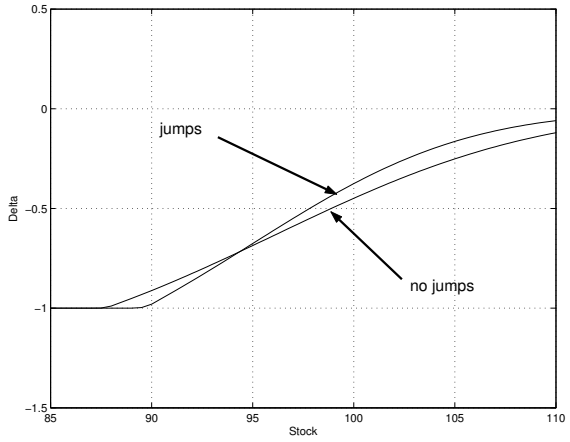
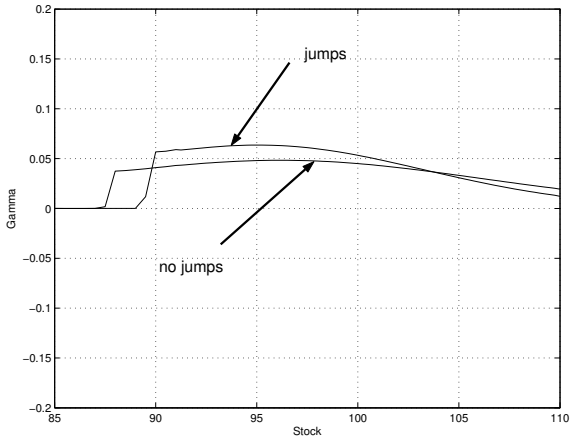


FIGURE 4.7: *The American put option value under the jump diffusion model is compared to the no-jump model (i.e. Black-Scholes). The no-jump model has an implied volatility which gives the same price as the jump model for a European option at the money. The input data is presented in Table 4.1. For the jump model the value of the American put option at the strike ($K = 100$) is $V = 3.24116$, while the value of the American put option under the no-jump model is $V = 3.25682$.*



(A) American put option delta.



(B) American put option gamma.

FIGURE 4.8: *American put option delta (V_S), and gamma (V_{SS}), jump diffusion model compared with model with no jumps. The no-jump model has an implied volatility which gives the same price as the jump model for a European option at the money. Data as in Table 4.1.*

American Butterfly Example

A more challenging numerical example is given by the solution to an American butterfly. Recall, a butterfly option has the payoff

$$V^* \quad = \quad \max(S - K_1, 0) - 2\max(S - (K_1 + K_2)/2, 0) + \max(S - K_2, 0) \ , \qquad (4.5)$$

(i.e. purchasing a low strike call K_1, selling two middle strike calls $(K_1 + K_2)/2$, and buying a high strike call K_2). This provides an interesting test case because the payoff pattern has a sharp kink at the middle strike. In our example, we assume the existence of an American style contract which specifies the payoff (4.5) ($K_1 = 90, K_2 = 110$), and we assume that the option can only be exercised early as a unit.

Table 4.8 shows a convergence study for the American butterfly. On each refinement, new nodes are inserted between each pair of coarse grid nodes and the timestep size is

approximately halved. Two timestepping methods were used. The implicit American constraint used Algorithm 3.3. The explicit American constraint used the following modification. Using the notation introduced in equation (3.64), we iterate for $\mathcal{V}^{k+1}$ (setting the penalty term to zero)

$$[I - \frac{L}{2}]\mathcal{V}^{k+1} = [I + \frac{L}{2}]V^n + \frac{\lambda\Delta\tau}{2}B\mathcal{V}^k + \frac{\lambda\Delta\tau}{2}BV^n. \tag{4.6}$$

After the iteration has converged, we then set

$$V^{n+1} = \max(V^*, \mathcal{V}^{k+1}) \ . \tag{4.7}$$

In this case, we would expect that the time truncation error is $O(\Delta\tau)$. In fact, this is clearly demonstrated in Table 4.8, since the ratio of changes appears to be asymptotically four for the implicit Crank-Nicolson American approach (which indicates quadratic convergence) compared to the asymptotic ratio of 2 (linear convergence) for the explicit American method. It is interesting to see from Table 4.8 that the number of iterations for the implicit American method is only slightly greater than for the explicit American technique. This indicates that we can impose the American constraint implicitly at very little computational expense compared to an explicit constraint method.

For comparison, we also show in Table 4.8 the results for a fully implicit discretization of the PDE term, an explicit evaluation of the correlation integral, and an explicit application of the American constraint. More precisely,

$$\begin{aligned} [I - L]\mathcal{V}^{n+1} &= V^n + \lambda\Delta\tau BV^n \\ V^{n+1} &= \max(V^*, \mathcal{V}^{n+1}) \ . \end{aligned} \tag{4.8}$$

This method is unconditionally stable (a straightforward extension of the proofs in

Table 4.8 Value of an American butterfly, $S = 105$, $t = 0$, under the jump diffusion process using the data presented in Table 4.1. American constraint applied implicitly is presented first. Algorithm (4.6-4.7): American constraint imposed explicitly. Algorithm (4.8): fully implicit PDE, explicit correlation integral, explicit American constraint. Itns is the total number of iterations required in algorithm (3.3), for all timesteps. Change is the change from one level of refinement to the next. Ratio is ratio of changes. Data as in Table 4.1. Crank-Nicolson timestepping is used with the timestep selector defined by (4.3), where dnorm = .05 and the initial timestep $\Delta t_{\text{init}} = .005$, on the coarsest grid. Grid and timestep sizes are reduced by half on each refinement.

Size of S grid	No. of Timesteps	Implicit American constraint			
		Itns	Value	Change	R
127	44	133	5.2490795		
254	111	249	5.2511148	.0020353	
508	246	546	5.2515158	.0004010	5.1
1016	511	1130	5.2515839	.0000689	5.8
2032	1042	2280	5.2516010	.0000171	4.0

Size of S grid	No. of Timesteps	Method (4.6-4.7)			
		Itns	Value	Change	R
127	43	129	5.2296200		
254	111	222	5.2429331	.0179502	
508	246	492	5.2475702	.0046371	3.9
1016	511	1022	5.2496169	.0020467	2.3
2032	1041	2082	5.2506144	.0009975	2.1

Size of S grid	No. of Timesteps	Method (4.8)			
		Itns	Value	Change	R
127	43	43	5.1997845		
254	111	111	5.2310288	.0312443	
508	246	246	5.2423694	.0113406	2.8
1016	511	511	5.2471667	.0047973	2.4
2932	1041	1041	5.2493929	.0022262	2.0

Theorem 3.1 [29] shows this), and is clearly the cheapest method (per timestep). It is also obvious that this method is consistent and monotone, and hence satisfies the sufficient

conditions for convergence to the viscosity solution [11, 21]. However, convergence is clearly only first order. As shown in [44], an explicit application of the American constraint can result in oscillations in gamma near the exercise boundary.

Figure 4.9 shows the value of an American butterfly, with the jump diffusion model (jumps) and the constant volatility Black-Scholes model (no-jumps). As described earlier, the constant volatility Black-Scholes model uses an implied volatility which reproduces the jump model price at $S = 100$ for a vanilla European call. The corresponding delta and gamma are shown in Figure 4.10.

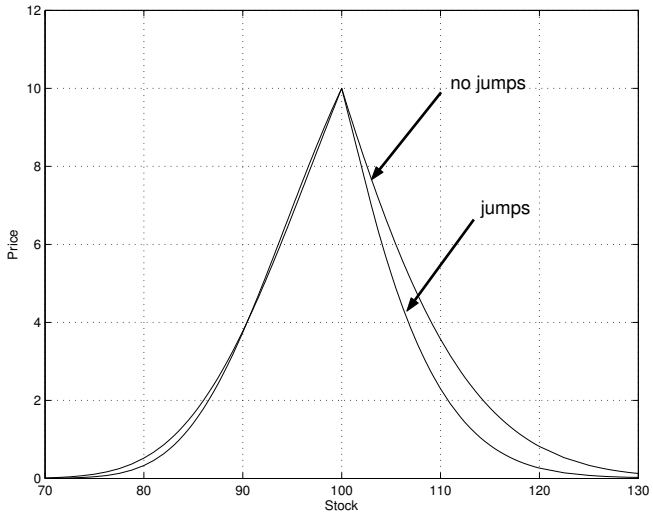
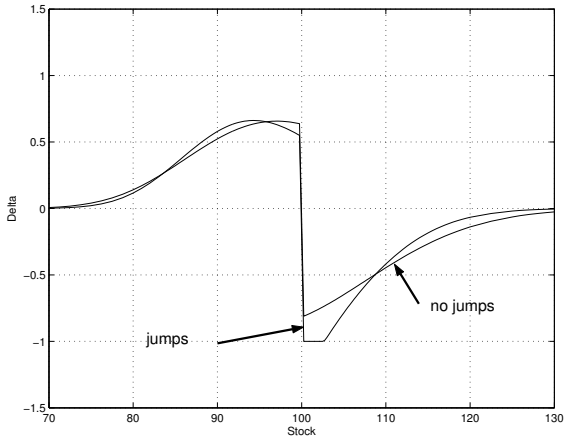
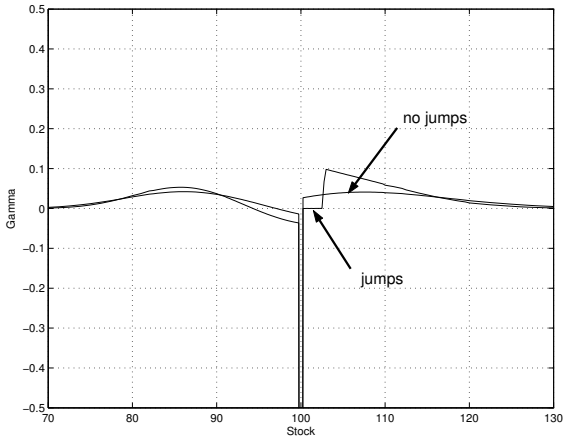


FIGURE 4.9: *American butterfly option value, jump diffusion model compared with model with no jumps. The no-jump model has an implied volatility which gives the same price as the jump model for a European option at the money. Data as in Table 4.1* $K_1 = 90, K_2 = 110$. *For the no-jump case the American put option is valued using* $\sigma = \sigma_{imp} = .1886$.

It is common in financial applications to impose the boundary condition $V_{SS} = 0, S \to \infty$. Consequently, we have repeated our numerical experiments using the boundary condition $V_{SS} = 0, S \to \infty$, which has the result that the coefficient matrices are no longer M-matrices and our convergence proofs no longer apply. In all cases the number of iterations required for convergence was virtually identical to the case where a Dirichlet condition was



(A) American put option delta.



(B) American put option gamma.

FIGURE 4.10: *American put option delta (V_S), and gamma (V_{SS}), jump diffusion model compared with model with no jumps. The no-jump model has an implied volatility which gives the same price as the jump model for a European option at the money. Data as in Table 4.1. For the no-jump case the American put option is valued using $\sigma = \sigma_{imp} = .1886$.*

imposed. The solutions also agreed up to six figures.

4.1.4 Exotic Options

The last set of results to be presented are for the case of a European call option with a Parisian knock-out feature. The particular case we consider here is an up-and-out call with daily discrete observation dates. This contract ceases to have value if S is above a specified barrier level for a specified number of consecutive monitoring dates. This can be valued by solving a set of one-dimensional problems which exchange information at monitoring dates [86]. It is easy to incorporate jumps by simply adding a jump integral term to each of the one-dimensional problems. Other path-dependent contracts such as Asian options can also be handled using this approach of solving a set of one-dimensional problems [95].

For our test, the barrier is set at $S = 120$ and the required number of consecutive daily observations for knock-out is 10. We consider the lognormal jump distribution case with the

same input parameters as in Table 4.1. Note that we specify the barrier observation interval as 1/250, based on 250 trading days per year. In Table 4.9, we present our convergence results. We use constant timestepping ($\Delta\tau = .002$ on the coarse grid) and the solution is l_2 projected after each barrier observation date. Rannacher timestepping is used after each observation. As expected, quadratic convergence is obtained.

Table 4.9 Value of an up-and-out Parisian call option using Rannacher timestepping with constant timesteps ($\Delta\tau = .002$ on the coarsest grid) and l_2 projection. The input parameters are given in Table 4.1. The barrier is set at $S = 120$ and 10 consecutive daily observations are required to knock-out. The convergence ratio R is defined in equation (4.2). The number of points used for the FFT grid is 2^α, where α is the smallest integer such that the number of nodes in the non-uniform S grid $p \leq 2^\alpha$. Quadratic interpolation is used.

Size of	No. of	$S = 90$		$S = 100$		$S = 110$	
S grid	Timesteps	Value	R	Value	R	Value	R
101	125	0.524766	n.a.	4.193418	n.a	8.762555	n.a.
201	250	0.523168	n.a	4.212131	n.a	8.779253	n.a.
401	500	0.522761	3.930	4.216747	4.053	8.782267	5.540
801	1000	0.522660	4.002	4.217902	3.997	8.783008	4.068
1601	2000	0.522634	4.015	4.218192	3.990	8.783199	3.875

In Figure 4.11, we compare the solutions of a Parisian call knock-out option with discrete daily observation dates with and without jumps.

We observe in Figure 4.11 that the difference in pricing can be significant for these parameter values, depending on the underlying asset price. The largest differences are near $S = 110$, where the model with jumps produces values of about 8.78 (as shown in Table 4.9), but the values for the no-jump model are around 7.25. For S ranging between about 98 and around 119, the jump model produces higher option values, but outside this range (in either direction) the model without jumps produces higher values.

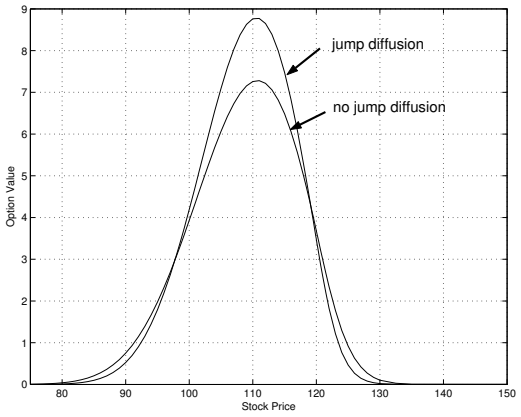


FIGURE 4.11: *Parisian knock-out call option with discrete daily observation dates with and without jumps. The barrier is set at* $S = 120$ *and the number of consecutive daily observations to knock-out is* 10*. In the no-jump case the Parisian knock-out call option is priced using* $\sigma = \sigma_{imp} = .1886$.

Table 4.10 Number of iterations for a European call option under jump diffusion using Crank-Nicolson timestepping. The input parameters are provided in Table 4.1. The convergence tolerance *tol* is defined in Algorithm 3.2.

Number of points N	Timesteps	Iterations ($tol = 10^{-6}$)	Iterations ($tol = 10^{-8}$)
128	25	77	100
255	50	150	200
509	100	300	390
1017	200	600	600
2033	400	1091	1200
4065	800	1600	2400

4.1.5 Remarks on Numerical Examples

It is worth concluding this section by making some comparisons with another method which have been proposed in the literature. When pricing options under the jump diffusion process, the main computational cost is the evaluation of the integral term of (3.5). The approach presented in [5] is based on a FFT-ADI finite difference method. This method evaluates the convolution integral twice at each timestep, thus requiring a total of four FFT computations (two forward FFTs, and two reverse FFTs). Note that the method in [5] is second order

accurate and unconditionally stable for European options. If N is the number of timesteps, and p the number of nodes in the S grid, then both the method in [5] and the method in this work have complexity $O(Np \log p)$.

In Table 4.10, we see that the number of iterations required for convergence (at each timestep) depends on the convergence tolerance. For a typical convergence tolerance of 10^{-6}, at most three iterations per step are required (on average). In this case, about six FFT computations are required per timestep. Consequently, for vanilla European options (with jumps), the method of [5] may be more efficient than the pure Crank-Nicolson timestepping method developed here. However, in the case of American options, it is not clear how the approach in [5] could be modified to handle the early exercise constraint implicitly, unless some form of iteration is used. In contrast, our technique can handle implicit treatment of the American constraint in a straightforward fashion.

4.2 Summary

The main results of this chapter are as follows:

- We showed that quadratic convergence was obtained when pricing a wide variety of financial options. Even if the initial conditions were not smooth (such as for digital options or Parisian options), quadratic convergence was recovered.

- We showed that for typical values of the timestep size and Poisson arrival intensity, the l_∞ error is reduced by two orders of magnitude at each iteration. On average three iterations per timesteps were enough for the fixed point algorithm to achieve convergence.

- We showed that when pricing American options under the jump diffusion process, the number of iterations per timesteps is relatively unchanged compared to pricing vanilla options under the jump diffusion process.

- In contrast with previous work, we did not assume that the grid was equally spaced in either the underlying asset price or its logarithm. Linear interpolation between the option grid and the fft grid did not affect the rate of convergence.

- Unlike previous work, the method developed here uses implicit timestepping for both the correlation integral term and the American constraint. As a result, we obtained higher order convergence (in terms of timestepping error) compared with previous methods which treat the correlation integral or the American constraint explicitly.

Chapter 5

Semi-Lagrangian Approach for Path Dependent Options

Recall equation (2.18) from Chapter 2, which gives the value of investment in telecommunication equipment:

$$V_t + \alpha(\eta - Q)V_Q + R + \lambda \int_0^\infty V(\eta, QJ, t)g(J)dJ + \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta} + \eta(\mu - \zeta\sigma)V_\eta - (r + \lambda)V = 0. \quad (5.1)$$

The objective of this thesis is to solve equation (5.1) to determine when it is optimal to add new capacity to existing telecommunication infrastructure. In the two previous chapters (Chapter 3 and 4), an efficient method was presented to handle the integral term in equation (5.1). However another important numerical issue must be dealt with before proceeding any further. Equation (5.1) is a two dimensional PIDE with no diffusion in the demand direction (Q). Classically, equations with no diffusion are solved using non-linear flux limiters [95, 94]. In this thesis, a semi-Lagrangian method is proposed as an alternative to solve equation (5.1). To validate our approach, we consider a two dimensional PDE taken from the financial

derivative pricing literature. A financial application (i.e. Asian option) was selected since the convective term becomes infinite as $\tau \to T$, hence this is an extremely difficult problem.

5.1 Introduction

An Asian option gives the holder a payoff that depends on the average price of the underlying asset over a specified period of time. Asian-style derivatives have a wide variety of applications in equity, energy, interest rate, and insurance markets. To the best of our knowledge, they were first introduced in [20]. For an historical review of Asian options, see [19].

Asian derivatives are very popular for several economic reasons. First, since the volatility of the average price is less than the volatility of the price, Asian options are less expensive than regular vanilla options. Second, while for some classes of derivatives it is possible for large market participants to manipulate the price of illiquid commodities, it is much harder to manipulate the average price over a period of time. Finally, companies are often more interested in the average price of oil or foreign exchange rate, than the underlying price or rate, when considering long term projects. For example, airline companies are certainly more interested in buying oil based on its average price instead of its spot price. Consequently, pricing Asian options accurately is critical.

The price of an Asian option at any time is a function of both the underlying asset at that time and the average of the underlying prices up to that time; as such these options are considered path-dependent. In practice, Asian option contracts specify that the average is monitored discretely. A typical situation would be to base the average on the daily closing price. In [34] it is shown that if daily averaging is used, then for typical market parameters, for options with expiry times more than three months, for all practical purposes (i.e. 1% accuracy) these options can be considered as being continuously monitored. In addition,

if we need to price long term (greater than one year) Asian options, then using timesteps of one day (which would be required in a discrete observation model [95]) would clearly be computationally wasteful. Consequently, in this chapter we focus on continuously observed Asian options. For details on numerical methods for discretely observed Asian options, we refer the readers to [28, 27, 95, 45].

In general, a two dimensional PDE must be solved to price an Asian option. In certain special cases, for example, constant volatility, no barrier features, and a floating strike contract, this problem can be reduced to a one-dimensional PDE [8]. However, in the general case (i.e. American style, asset dependent barriers or volatility), the two dimensional PDE cannot be reduced to one dimension. This PDE is difficult to solve, since the pricing equation has no diffusion in one of the coordinate directions. In [94, 96], a non-linear flux limiter was used to retain accuracy while preventing oscillations. In [62], the first order hyperbolic term was discretized using a first order upwind type method, resulting in at most first order accuracy. In [69], a semi-Lagrangian (SL) method was used to discretize the hyperbolic term in the average direction. Semi-Lagrangian schemes were first introduced by [36] and [72] for atmospheric and weather numerical predictions. These are time marching schemes that integrate convection-diffusion equations by tracing the characteristic backward in time. These schemes are used to reduce numerical issues raised by convection dominated equations. In principle, provided an appropriate time discretization and a high enough order of interpolation to recover values at the feet of the characteristic curves [43, 2, 16], then SL methods are able to have greater than first order convergence as the grid and the timestep size are reduced. As such SL schemes can provide much more accurate solution when comparing to the more traditional methods for pricing Asian options where high order timestepping schemes (e.g. Crank-Nicolson, second order backward differencing) are used but the jump condition is applied explicitly. For example, this standard approach was demonstrated in [95], and only first order (in time) convergence was obtained. The

semi-Lagrangian approach is particularly attractive, since only a small change to existing software (which uses the traditional approach of an explicit jump condition) is required to obtain higher oder convergence.

5.2 Mathematical Model

Let S represent the underlying stock price. Potential paths followed by the stock can be modeled by the stochastic process given by equation (3.1).

When the average is monitored continuously [12, 89, 45], the arithmetic average A is defined as

$$\begin{aligned} A &= \frac{\int_0^t S(u)du}{t}, \\ dA &= \frac{(S-A)dt}{t}. \end{aligned} \tag{5.2}$$

Using standard arguments [12], the value of an option depending on the average A (5.2) and S (3.1) is given by

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + \frac{(S-A)}{t}V_A + (r-\lambda\kappa)SV_S - (r+\lambda)V + \lambda\int_0^\infty V(SJ)g(J)dJ = 0, \tag{5.3}$$

where r is the continuously compounded risk free interest rate. Since we are solving backward in time from the expiration time $t = T$ to the present time $t = 0$, equation (5.3) becomes

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + \frac{(S-A)}{T-\tau}V_A + (r-\lambda\kappa)SV_S - (r+\lambda)V + \lambda\int_0^\infty V(SJ)g(J)dJ, \tag{5.4}$$

where $\tau = T - t$. It is important to note that equation (5.4) has no diffusion term in the A direction and this is the source of many numerical difficulties [94].

In order to completely specify our problem, boundary conditions for equation (5.4) must be specified. For the terminal boundary conditions, a number of common payoffs for pricing different types of Asian securities can be used. Typical examples include

- fixed strike call: $V(S, A, \tau = 0) = \max(A - K, 0)$,
- fixed strike put: $V(S, A, \tau = 0) = \max(K - A, 0)$.

For the non-terminal boundary conditions, at $S = 0$ equation (5.4) reduces to

$$V_\tau = \frac{-A}{T - \tau} V_A - rV. \tag{5.5}$$

Note that no boundary conditions are required at $A = 0$ or as $A \to \infty$, since the characteristics of the PIDE are outgoing in the A direction. The boundary condition at $S \to \infty$ is, however, more difficult to specify and requires additional justification. If we make the common assumption that $V_{SS} \to 0$ as $S \to \infty$, then this implies that

$$V \simeq f(A, \tau)S + g(A, \tau) \tag{5.6}$$

as $S \to \infty$ which then means that equation (5.4) becomes

$$V_\tau = \frac{S - A}{T - \tau} V_A + rSV_S - rV. \tag{5.7}$$

For A very large and $S \to \infty$, we can approximate the solution to equation (5.7) (see [68] for details) by

$$V \simeq H_1(\tau)A + H_2(\tau)S + H_3(\tau) \tag{5.8}$$

so that

$$V \simeq \frac{D_1}{T} e^{-r\tau}(T-\tau)A + \left[\frac{D_1}{rT}(1-e^{-r\tau}) + D_2\right] S + D_3 e^{-r\tau} \tag{5.9}$$

where D_1, D_2, D_3 are independent of (S, A, τ) and are determined by the payoff. For example, for a fixed strike call, $D_1 = 1, D_2 = 0, D_3 = -K$. We then use equation (5.9) at all points on $S = S_{\max}$ (which is clearly an approximation for A small) so that

$$V_S \simeq \left[\frac{D_1}{rT}(1-e^{-r\tau}) + D_2\right]; S \to \infty. \tag{5.10}$$

Substituting equation (5.10) into equation (5.7) gives

$$V_\tau = \left(\frac{S-A}{T-\tau} V_A + \upsilon(S,\tau) - rV\right); S \to \infty, \tag{5.11}$$

where

$$\upsilon(S,\tau) = \left[\frac{D_1}{rT}(1-e^{-r\tau}) + D_2\right] rS. \tag{5.12}$$

The use of approximation (5.10) is discussed in [68], where it is mentioned that estimate (5.10) is in fact an upper bound for V_S. It must be admitted that use of equation (5.9) at all points along $S = S_{\max}$ is not rigorously justified. However, we note that other authors [62] simply specify that the boundary condition at $S = S_{\max}$ is set to the payoff. In [62], the size of the computational domain is increased as the grid size is reduced, so that the effect of poor specification of the boundary condition becomes negligible. In any case, some numerical experiments are included with varying choice of $S_{\max}$ to show that in practice this specification is not critical.

In practice, we solve for the solution of equation (3.61) on the finite computational

domain $[0, S_{\max}] \times [0, A_{\max}]$. Usually $S_{\max} = A_{\max}$. If $S_{\max}$ is sufficiently large, then the errors introduced by imposing condition (5.11) are small in regions of interest. This will be verified in some numerical experiments.

For the payoffs mentioned earlier, the corresponding formulas for $\upsilon(\tau, S)$ are given by

- fixed strike call: $\upsilon(S, \tau) = \frac{(1-e^{-r\tau})}{T} S$
- fixed strike put: $\upsilon(S, \tau) = 0$.

5.3 Semi-Lagrangian Discretization

This section explores different discretization methods for the partial differential equation using the semi-Lagrangian approach. Before proceeding any further, let us introduce the following definitions. We use an unequally spaced grid in S coordinates for the PDE discretization $[S_0, \ldots, S_{imax}]$, and similarly use an unequally spaced grid in the A direction $[A_0, \ldots A_{jmax}]$. Let

$$V_{i,j}^n = V(S_i, A_j, \tau^n) \tag{5.13}$$

denote the solution at asset price node S_i for the average A_j and time level n. Let $\mathcal{L}$ be the differential operator represented by

$$\mathcal{L}V \equiv \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa) S V_S - (r + \lambda) V, \tag{5.14}$$

and

$$\mathcal{I}V \equiv \lambda \int_0^\infty V(SJ) g(J) dJ. \tag{5.15}$$

Equation (5.4) can then be rewritten as

$$V_\tau - \frac{(S - A)}{T - \tau} V_A = \mathcal{L}V + \mathcal{I}V. \tag{5.16}$$

We use standard finite difference methods to discretize the operator $\mathcal{L}V$ (see Chapter 3 and [73]). If we impose boundary condition (5.11), and use forward and backward differencing as appropriate, it is easy to see that the discrete form of $[I - \mathcal{L}V]_i$ (where I is the identity matrix) is an M-matrix. As discussed in [73], for typical values of σ, r, upwind differencing of the V_S term in equation (5.16) is required only rarely, and usually remote from regions of interest, so that in practice this does not impact solution quality. Requiring the discrete form of $\mathcal{L}V$ to be an M-matrix has interesting theoretical properties. In the following, we denote the discrete form of $\mathcal{L}V$ at $S = S_i, A = A_j, \tau = \tau^n$ by $(\mathcal{L}V)^n_{i,j}$.

Keeping S constant, the Lagrangian derivative along a trajectory $A = A(S, \tau)$ is

$$\frac{DV}{D\tau} = \frac{\partial V}{\partial \tau} + \frac{\partial V}{\partial A}\frac{dA}{d\tau} . \tag{5.17}$$

Along the trajectory

$$\frac{dA}{d\tau} = \frac{A - S}{T - \tau} \tag{5.18}$$

equation (5.16) can be written as written as

$$\frac{DV}{D\tau} = \mathcal{L}V + \mathcal{I}V. \tag{5.19}$$

Let $A = A(S_i, A_j, \tau^{n+1}, \tau)$ be a trajectory satisfying equation (5.18), which passes through the discrete grid point (S_i, A_j) at $\tau = \tau^{n+1}$. Let $V^n_{i,j(i,n+1)} = V(S_i, A^n_{j(i,n+1)}, \tau^n)$ be the value of the option price at the feet $A^n_{j(i,n+1)} = A(S_i, A_j, \tau^{n+1}, \tau^n)$ of the characteristic curve defined by equation (5.18). Note that $A^n_{j(i,n+1)}$ will not necessarily coincide with a

grid point A_j. $A^n_{j(i,n+1)}$ is determined by solving

$$\begin{aligned} \frac{dA}{d\tau} &= \frac{A - S_i}{T - \tau}, \\ & A = A_j \ ; \ \tau = \tau^{n+1}, \\ & A^n_{j(i,n+1)} = A^{n+1}_j + \int_{\tau^{n+1}}^{\tau^n} \left(\frac{dA}{d\tau} \right) d\tau \ . \end{aligned} \tag{5.20}$$

from $\tau = \tau^{n+1}$ to $\tau = \tau^n$. In general, the integration in equation (5.20) must be computed numerically and the integration method should have an error one order higher than the timestepping method used to approximate the Lagrangian derivative [2]. Figure 5.1 graphically presents how the value $V^n_{i,j(i,n+1)}$ is found by tracing back along the characteristic path $A = A(S_i, A_j, \tau^{n+1}, \tau)$.

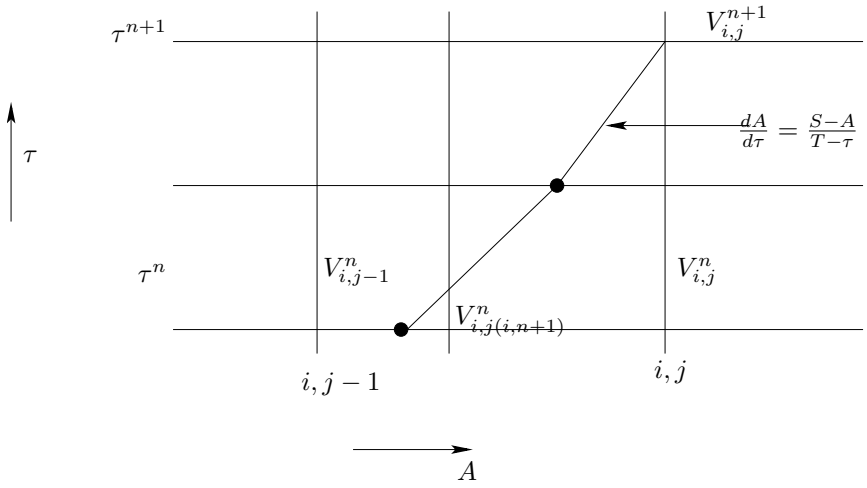


FIGURE 5.1: *The value of $V^n_{i,j(i,n+1)} = V(S_i, A^n_{j(i,n+1)}, \tau^n)$ is traced back along the characteristic path $A = A(S_i, A_j, \tau^{n+1}, \tau)$. $V(S_i, A^n_{j(i,n+1)}, \tau^n)$ is found by interpolation along the A direction.*

Discretizing equation (5.19) along the characteristic trajectory for different timestepping schemes gives, in the case of fully implicit timestepping:

$$\frac{V^{n+1}_{i,j} - V^n_{i,j(i,n+1)}}{\Delta\tau} = (\mathcal{L}V)^{n+1}_{i,j} + (\mathcal{I}V)^{n+1}_{i,j}, \tag{5.21}$$

for Crank-Nicolson timestepping (CN),

$$\frac{V_{i,j}^{n+1} - V_{i,j(i,n+1)}^{n}}{\Delta\tau} = \frac{1}{2}\left[(\mathcal{L}V)_{i,j}^{n+1} + (\mathcal{I}V)_{i,j}^{n+1}\right] + \frac{1}{2}\left[(\mathcal{L}V)_{i,j(i,n+1)}^{n} + (\mathcal{I}V)_{i,j(i,n+1)}^{n}\right], \quad (5.22)$$

and for second order backward differencing (BDF) [15]

$$\frac{\frac{1}{2}V_{i,j}^{n+1} - 2V_{i,j(i,n+1)}^{n} + \frac{1}{2}V_{i,j(i,n+1)}^{n-1}}{\Delta\tau} = (\mathcal{L}V)_{i,j}^{n+1} + (\mathcal{I}V)_{i,j}^{n+1}. \quad (5.23)$$

For ease of exposition, we have written equation (5.23) for constant timesteps. This is trivially generalized to non-constant timesteps [15].

Unlike traditional applications of the semi-Lagrangian approach where the characteristic curve must be estimated numerically, for Asian options the solution along the characteristic curve can be determined exactly. Regarding S as a constant, and solving equation (5.18) gives

$$A = S_i + \frac{C}{T - \tau}, \quad (5.24)$$

where C is a constant. At $\tau = \tau^{n+1}$, $A = A_j^{n+1}$, so that

$$\begin{aligned} \text{At time } \tau^{n} \quad &: \quad A_{j(i,n+1)}^{n} = A_j^{n+1} + \frac{(S_i - A_j^{n+1})(\tau^{n+1} - \tau^{n})}{T - \tau^{n}}, \\ \text{At time } \tau^{n-1} \quad &: \quad A_{j(i,n+1)}^{n-1} = A_j^{n+1} + \frac{(S_i - A_j^{n+1})(\tau^{n+1} - \tau^{n-1})}{T - \tau^{n-1}}, \end{aligned} \quad (5.25)$$

where $T \geq \tau^{n+1} > \tau^{n} > \tau^{n-1}$. It is interesting to observe that for the last step when $\tau^{n+1} = T$, equation (5.25) simplifies to $A_{j(i,n+1)}^{n} = A_{j(i,n+1)}^{n-1} = S_i$.

The various quantities $(\cdot)_{i,j(i,n+1)}^{n}$ in equations (5.21-5.23) are determined by interpolation along lines of constant $S = S_i$. Assuming that the S derivatives and the integral term are discretized using second order accurate methods, then it follows from [16] that at least quadratic interpolation should be used for $(\cdot)_{i,j(i,n+1)}^{n}$ so as to retain global second order

convergence.

5.4 Semi-Lagrangian Timestepping and Discrete Observation

It is common to consider continuously observed Asian options as the limit of discretely observed Asian options as the observation interval tends to zero [28, 84]. In this section, we show that if the discrete sampling period is equal to the discrete PDE timestep, then a fully implicit, discretely sampled model is algebraically identical to a fully implicit semi-Lagrangian discretization of a continuously observed model. In the following, for simplicity the effect of the boundary condition (5.11) are ignored.

Consider the discrete average computed at discrete averaging times $t^l = l\Delta t$.

$$A(t^l) = \frac{1}{k}\sum_{k=1}^{k=l} S(t^k). \tag{5.26}$$

Equation (5.26) can be written

$$A(t^{l+1}) = A(t^l) + \frac{S(t^{l+1}) - A(t^l)}{l+1}. \tag{5.27}$$

Now, when using a PDE method to price a discretely observed Asian option, we consider that $V = V(S, A, t)$, and regard (S, A) as independent variables. Suppose we have N observation dates, at the times $\Delta t, 2\Delta t, ..., N\Delta t = T$. Then at the $l'th$ observation date we must have, by no arbitrage [89]

$$\begin{aligned} V(S, A(t^{l+1}), t^{(l+1)+}) &= V(S, A(t^l), t^{(l+1)-}), \\ A(t^{l+1}) &= A(t^l) + \frac{S - A(t^l)}{l+1}, \end{aligned} \tag{5.28}$$

where $t^{(l+1)+}, t^{(l+1)-}$ are the instants just after and before the observation date t^{l+1}. Note

that $A(t^{l+1})$ is regarded as constant for $t^{l+1} < t < t^{l+2}$. Now let $l = N - k$, so that k counts backwards. Note that $\tau^k = k\Delta\tau$ and that

$$t^l = l\Delta t = (N-k)\Delta t = T - (k\Delta\tau) = \tau^k, \tag{5.29}$$

and similarly $t^{l+1} = \tau^{k-1}$. As well, we have that $t^{(l+1)+} = \tau^{(k-1)-}$, $t^{(l+1)-} = \tau^{(k-1)+}$, and writing jump condition (5.28) in terms of $\tau = T - t$, gives

$$\begin{aligned} V(S, A(\tau^k), \tau^{k-}) &= V(S, A(\tau^{k+1}), \tau^{k+}), \\ & \quad A(\tau^k) = A(\tau^{k+1}) + \frac{S - A(\tau^{k+1})}{N-k}. \end{aligned} \tag{5.30}$$

Note that in this case we regard $A(\tau^{k+1})$ as fixed during $\tau^k < \tau < \tau^{k+1}$.

Consider the case of a discretely observed European Asian option. In this case we solve

$$V_\tau = \mathcal{L}V + \mathcal{I}V \tag{5.31}$$

on the domain $[0, S_{imax}] \times [0, A_{jmax}]$, with the jump conditions (5.30) imposed at observation times. Away from observation dates, if we discretize equation (5.31) in the A direction, then equation (5.31) represents a set of one dimensional PIDEs, which communicate only through jump conditions [28].

We can write the jump condition (5.30) as

$$\begin{aligned} V(S, A^k, \tau^{k-}) &= V(S, A^{k+1}, \tau^{k+}) \\ & \quad A^k = A^{k+1} + \frac{(S - A^{k+1})\Delta\tau}{T - \tau^k} \end{aligned} \tag{5.32}$$

where we have used the notation $A(\tau^k) = A^k$.

Recalling that A^{k+1} is constant during $\tau^{k+} \leq \tau \leq \tau^{(k+1)-}$ then a fully implicit dis-

cretization of equation (5.31) gives

$$\frac{V(S_i, A_j^{k+1}, \tau^{(k+1)-}) - V(S_i, A_j^{k+1}, \tau^{k+})}{\Delta\tau} = (\mathcal{L}V)_{i,j}^{k+1} + (\mathcal{I}V)_{i,j}^{k+1}. \quad (5.33)$$

Note that this is a set of independent one dimensional PDEs (there are no A derivatives in equation (5.33), A_j^{k+1} appears only as a parameter). Using the jump condition (5.32) in equation (5.33) gives

$$\begin{aligned} \frac{V(S_i, A_j^{k+1}, \tau^{(k+1)-}) - V(S_i, A_{j(i,k+1)}^{k}, \tau^{k-})}{\Delta\tau} &= (\mathcal{L}V)_{i,j}^{k+1} + (\mathcal{I}V)_{i,j}^{k+1} \\ A_{j(i,k+1)}^{k} &= A_j^{k+1} + \frac{(S - A_j^{k+1})\Delta\tau}{T - \tau^k} \end{aligned} \quad (5.34)$$

which we recognize from equation (5.21) and equation (5.25) as being algebraically identical to a semi-Lagrangian, fully implicit discretization.

Note that in order for this result to hold, we must have discrete observations at $t = \Delta t, 2\Delta t, ..., N\Delta t$, i.e. no observation at $t = 0$. Of course, in the limit as $\Delta t \to 0$, adding an extra observation at $t = 0$ will be the same to $O(\Delta t)$ as the semi-Lagrangian solution.

Remark 5.1. *As discussed in [95], it is straightforward to show that the common lattice methods used to price Asian options [53] are simply explicit finite difference methods for discretely observed models of Asian options. In many lattice applications, the observation interval is set to the lattice timestep, hence the continuously observed price is computed in the limit of vanishing timestep. A straightforward extension of the results above can be used to show that these lattice methods are simply explicit semi-Lagrangian methods. In this case, it is also easy to derive the conditions on the order of interpolation and the spacing on the lattice in the average direction to ensure optimal convergence. We note that, as discussed in [95], this is a point of confusion in the finance literature, and has led to schemes which are not, in fact, convergent [12].*

5.5 Properties of the Discrete Equations

In the following, the properties of the discrete equations are investigated. We first, note that the algebraic equations (5.34) are decoupled for each line of constant A_j at each timestep, resulting in a set of one dimensional discrete PIDEs. Hence we can use the techniques in Chapter 3 and in [44, 73, 29, 32] to prove certain properties of the discretized equations. In the following we give sketches of the proofs of these properties.

5.5.1 Preliminaries

Define the matrix L such that

$$\left[L \cdot V_j^n\right]_i = \Delta\tau \left(\mathcal{L}V\right)_{i,j}^n + \text{ interpolation error}, \tag{5.35}$$

and

$$\lambda \left[B \cdot V_j^n\right]_i = \left(\mathcal{I}V\right)_{i,j}^n + \text{ interpolation error}, \tag{5.36}$$

where V_j^n is the vector of discrete solution values $\left[V_j^n\right]_i = V(A_j, S_i, \tau^n)$ for fixed A_j. A detailed description of B and L are given in Chapter 3 equations (3.51) and (3.52). To avoid algebraic complication, the discrete equations, and the method used to solve the algebraic equations, are described only for the fully implicit and Crank-Nicolson timestepping methods. The reader should have no difficulty generalizing the results to the BDF case.

Let Φ^{n+1} be the Lagrange interpolation operator such that

$$\left(\Phi^{n+1} \cdot V^n\right)_{i,j} = V(S_i, A^n_{j(i,n+1)}, \tau^n) + \text{ interpolation error} \tag{5.37}$$

where Φ^{n+1} is a linear operator for any order (linear, quadratic) of interpolation. Then the

matrix form of the discrete equations is given by

$$[I-(1-\theta)L]V_j^{n+1} = \left[\Phi^{n+1}[I+\theta L+\theta\lambda\Delta\tau B]V^n\right]_j + (1-\theta)\lambda\Delta\tau BV_j^{n+1} + \Delta\tau F_j^{n+1} \quad (5.38)$$

for $j = 1, \ldots, jmax$. Here $\theta = 0$ is fully implicit, and $\theta = 1/2$ is Crank-Nicolson timestepping. The term F_j^{n+1} is used to approximate the boundary condition at $S = S_{imax}$. The boundary condition is also enforced at $i = imax$ by setting

$$L_{imax,l} = 0 \text{ and } B_{imax,l} = 0 \quad (5.39)$$

for $l = 1, \ldots, jmax$ and letting

$$\left[F_j^{n+1}\right]_i = \begin{cases} 0 & , i \neq imax \\ (1-\theta)\upsilon(S_i, \tau^{n+1}) + \theta\upsilon(S_i, \tau^n) & , i = imax \end{cases} \quad (5.40)$$

where $\upsilon(S, \tau)$ is defined in equation (5.12).

The choice of interpolation scheme is discussed in [45] and [43]. Specifically, if the interpolation error does not get damped out, the global interpolation error after N timesteps is $O\left(\frac{(\Delta S_{\max})^q}{\Delta\tau}\right)$, where $q = 2$ for linear interpolation, $q = 3$ for quadratic interpolation and $\Delta S_{\max} = \max_i(S_{i+1} - S_i)$. Assuming second order in space and time truncation errors, the global discretization error is

$$\text{global discretization error} = O\left[\frac{(\Delta S_{\max})^q}{\Delta\tau} + (\Delta S_{\max})^2 + (\Delta\tau)^2\right], \quad (5.41)$$

If we assume $\Delta S_{\max} = const.h$ and $\Delta\tau = const.h$, then equation (5.41) reduces to

$$\text{global discretization error} = O\left[\min((h^{q-1}, h^2)\right] \quad (5.42)$$

5.5.2 Monotonicity, Stability and Convergence of the Nonlinear Iteration

As in Chapter 4, we may have non-smooth solutions to equation (5.4). This may be due to the degeneracy of the diffusion operator (no diffusion in the A direction), and due to possible non-smoothness in the payoff function. In these cases, we seek the viscosity solution to equation (5.4) [11]. In the following we investigate the properties of equation (5.38) concentrating on monotonicity and stability as in Chapter 4.

It is convenient to gather together a set of conditions required for the following results:

Conditions 5.1.

- $[I - L]$ *is an* M*-matrix [equation* (5.38)*];*
- $[B \cdot V_j^n]_i \simeq \sum_k b_{ik} V_{kj}$ *[equation* (5.35)*],* $\quad 0 \leq b_{ik} \leq 1 \quad$ *and* $\sum_k b_{ik} \leq 1$*;*
- *Linear interpolation is used [equation* (5.37)*];*
- *Boundary conditions* (5.39) *and* (5.40) *are employed.*

Remark 5.2. *These conditions are satisfied if we use the discretization methods presented in Chapter 3.*

Lemma 5.1 (Stability of the fully implicit scheme ($\theta = 0$, equation (5.38)**)).** *The discretization method* (5.38) *is unconditionally stable provided that conditions 5.1 are satisfied.*

Proof. The use of semi-Lagrangian timestepping decouples the discrete equation at each timestep into a set of one dimensional discrete PIDEs. Consequently, using similar arguments as in Theorem 3.1, Chapter 3, it follows that equation (5.38) for $\theta = 0$ is unconditionally stable. □

Lemma 5.2 (Monotonocity of the discretization). *The fully implicit discretization* (5.38) *is monotone provided that conditions 5.1 are satisfied.*

Proof. This proof is a trivial extension of the proof of Lemma 3.1, Chapter 3, and is omitted. □

Remark 5.3 (Extension to nonlinear models). *It is completely straightforward to include a transaction cost or uncertain volatility model in the basic option pricing PDE [75], which makes the PDE nonlinear (even in the European case). For example, using the methods in [75], it is a simple exercise to extend the above stability and monotonicity results to the case of an American Asian option, with jumps and transaction costs [30].*

Theorem 5.1 (Iterative solution of the discretized equations). *The nonlinear algebraic equations* (5.38) *are solved using Algorithm 5.1. Provided that the conditions 5.1 are satisfied, then the fixed point iteration Algorithm 5.1 is globally convergent.*

Proof. The discrete equations (5.38) are decoupled for each line of constant A_j. Hence the issue of convergence of Algorithm 5.1 reduces to the convergence of each set of equations for constant j, hence the results follows from Theorem 3.3 in Chapter 3. □

Algorithm 5.1 Fixed point iteration scheme to price Asian option using a semi-Lagrangian approach.

Let $(V_j^{n+1})^0 = V_j^n$
Let $\hat{V}_j^{\,k} = (V_j^{n+1})^k$
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Solve
  $[I - (1-\theta)L]\,\hat{V}_j^{\,k+1} = \left[\phi^{n+1}\left[I + \theta L + \theta\lambda\Delta\tau B\right]V^n\right]_j + (1-\theta)\lambda\Delta\tau B\hat{V}_j^{\,k} + \Delta\tau F_j^{n+1}$
  if $\max_i \frac{|\hat{V}_{i,j}^{k+1} - \hat{V}_{i,j}^{k}|}{\max(1, |\hat{V}_{i,j}^{k+1}|)} < tolerance$ **then**
    Quit
  end if
end for

Remark 5.4 (Monotonicity of Crank-Nicolson ($\theta = \frac{1}{2}$)). *Crank-Nicolson is only conditionally monotone, hence only conditionally satisfies the sufficient conditions in [11, 21].*

However, if the Rannacher modification of Crank-Nicolson is used, then, consistent with the results in [73], our experimental computations indicate that fully implicit and Crank-Nicolson timestepping converge to the same solution. We conjecture that convergence to the viscosity solution can be obtained under somewhat weaker conditions than strict monotonicity as suggested in [11].

5.6 Computational Details and Numerical Results

This section presents numerical results for various options and payoffs. We begin our presentation of results by an analysis of the convergence rate of a continuously observed fixed strike Asian option with no jump ($\lambda = 0$). We use an unequally spaced grid in the A, S directions, on the domain $[0, S_{imax}] \times [0, A_{jmax}]$, with $A_{jmax} = S_{imax}$ (details regarding the grids that are used in this Chapter can be found in Appendix I). Probabilistic arguments can be used to determine an appropriate value for S_{imax} [84]. We use $S_{imax} = 50K$, where K is the strike. We describe below some tests which were carried out to verify that the effect of imposing boundary conditions at $S = 50K$ results in insignificant error. The convergence tolerance in iteration (3.3) was $tol = 10^{-6}$. If Crank-Nicolson or BDF timestepping is used, then quadratic interpolation is used in equation (5.37). If fully implicit timestepping is employed, then linear interpolation is used in equation (5.37).

Given an A grid discretization, the discrete PDEs (5.21-5.22) become decoupled. At each timestep, we have a set of independent one dimensional discrete PDEs to solve. This property makes solution of the continuously observed Asian option straightforward to implement, given an existing library which supports pricing of path dependent options.

As pointed in equation (5.42), it is necessary to use at least a quadratic Lagrange interpolation scheme to find the solution along the characteristic curve, if we hope to obtain quadratic convergence. This will, however, result in a scheme which is not monotone. The

convergence ratio R is defined by equation (4.2).

Table 5.1 Value of a continuously observed fixed strike European Asian call option (no jumps) with constant timesteps. The input parameters are $\sigma = .1$, $r = .1$, $T = .25$ and $K = 100$. We compare the results given using the Večeř [87] one dimensional model, and the semi-Lagrangian method presented here. Crank-Nicolson timestepping was used.

semi-Lagrangian				Večeř 1-D PDE [87]			
Size of S and A grids	No. of timesteps	$S = 100$ Value	R	Size of S grids	No. of timesteps	$S = 100$ Value	R
51	25	1.857193	n.a.	51	25	1.839863	n.a.
101	50	1.853254	n.a.	101	50	1.848642	n.a
201	100	1.852120	3.475	201	100	1.850851	3.974
401	200	1.851781	3.338	401	200	1.851407	3.979
801	400	1.851660	2.815	801	400	1.851546	3.987

Table 5.1 shows results for a low volatility case, European Asian option (no jumps), using the semi-Lagrangian approach. In this special case, the two dimensional PDE can be reduced to one dimension [87], which we refer to as the Večeř PDE [87] in the following. Results obtained by solving the Večeř PDE numerically are also given in Table 5.1.

In Table 5.1, we observe that the convergence ratio R for the semi-Lagrangian method is not quadratic ($R \neq 4$), while for the Večeř PDE [87] quadratic convergence is found. As discussed in [87], the Večeř PDE is not convection dominated, hence it is straightforward to obtain accurate numerical solutions. We remind the reader that this clever reduction to one dimension cannot be used for American options. The non-smooth payoff greatly affects the convergence of the semi-Lagrangian method. There is very little diffusion in the A direction, and the payoff non-smoothness at the strike is not smoothed out during the solution phase. It is probable that since quadratic interpolation is used in the A direction in order to determine the values of the solution at the feet of the characteristic curves (see Figure 5.1), the interpolation may be affected by the non-smooth payoff, and may lower the observed rate of convergence.

Next the effect of the boundary condition (5.11) at $S = S_{imax}$ is considered. Two different values for S_{imax} are used and the results are presented in Table 5.2. This table seems to indicate that there is a negligible error for options of this maturity incurred by setting $S_{imax} = 50K$. All subsequent results will be reported imposing condition (5.11) at $S_{imax} = 50K$.

Table 5.2 Value of a continuously observed fixed strike Asian call option at $S = K = 100$, constant Crank-Nicolson timestepping. The input parameters are $\sigma = .1$, $r = .1$, $T = .25$, $K = 100$. Convergence ratios (4.2) are presented for different timestepping schemes. The right boundary of the space discretization $[0, S_{imax}]$ domain is truncated at different values.

	$S_{imax} = 5 \times K$		$S_{imax} = 50 \times K$	
Timesteps	S, A grid nodes	Value	S, A grid nodes	Value
25	51	1.857193	54	1.857193
50	101	1.853254	109	1.853254
100	201	1.852120	217	1.852120
200	401	1.851781	433	1.851781
400	801	1.851660	865	1.851660

Figures 5.2 and 5.3 graphically present the solution V and the first derivative of the solution with respect to the stock price V_S when Crank-Nicolson is used. The plots are all smooth and do not exhibit any oscillations. While not shown here, V_{SS} also did not show any oscillations.

We now explore numerical convergence for pricing Asian options for large values of volatility σ. Table 5.3 presents our results. As expected quadratic convergence is recovered. In this case, a sufficient amount of diffusion in the S direction appears to compensate the little diffusion in the A direction.

5.6.1 An In Depth Study of the Convergence Ratio

The results of the previous section indicated that the semi-Lagrangian approach, coupled with Crank-Nicolson timestepping, results in quadratic convergence for large volatilities.

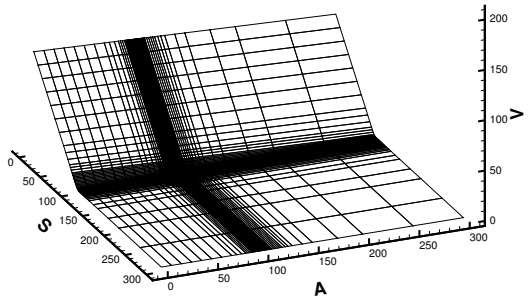


FIGURE 5.2: *Value of a European fixed strike Asian put using Crank-Nicolson with constant timestepping ($\Delta\tau = .01$). 51 grid points are used both in the A and S direction. The input parameters are $\sigma = .1$, $r = .1$, $T = .25$, and $K = 100$.*

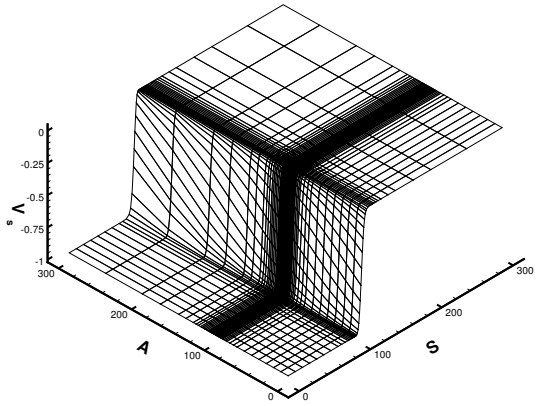


FIGURE 5.3: *First derivative value of a European fixed strike Asian put using Crank-Nicolson with constant timestepping ($\Delta\tau = .01$). 51 grid points are used both in the A and S direction. The input parameters are $\sigma = .1$, $r = .1$, $T = .25$, and $K = 100$.*

Table 5.3 Value of a continuously observed fixed strike Asian call option with constant timesteps at $S = K$. The input parameters are $\sigma = .5$, $r = .05$, $T = .25$ and $K = 100$. We compare the results given using the Večeř 1-D PDE [87], and the semi-Lagrangian method presented here. Crank-Nicolson timestepping was used.

semi-Lagrangian				Večeř 1-D PDE [87]			
Size of	No. of	$S = 100$		Size of	No. of	$S = 100$	
S and A grids	timesteps	Value	R	S grids	timesteps	Value	R
51	25	6.010203	n.a.	51	25	6.009821	n.a.
101	50	6.015092	n.a.	101	50	6.014848	n.a
201	100	6.016344	3.905	201	100	6.016251	3.582
401	200	6.016651	4.085	401	200	6.016619	3.816
801	400	6.016723	4.219	801	400	6.016713	3.915

But, for small volatility values, quadratic convergence was not recovered. The goal of this section is to explore in detail different numerical techniques that could improve the convergence rate.

Table 5.4 Value of a continuously observed fixed strike Asian call option at the strike, constant timesteps. The input parameters are $\sigma = .1$, $r = .1$, $T = .25$, $K = 100$. Convergence ratios (4.2) are presented for different timestepping schemes: implicit, Crank-Nicolson and second order BDF.

		Implicit timestepping		CN timestepping		BDF timestepping	
Size of	No. of	$S = 100$		$S = 100$		$S = 100$	
S and A grids	timesteps	Value	R	Value	R	Value	R
51	25	1.911865	n.a.	1.857193	n.a	1.86096	n.a.
101	50	1.880801	n.a.	1.853254	n.a	1.854310	n.a.
201	100	1.865907	2.086	1.852120	3.475	1.852416	3.513
401	200	1.858681	2.061	1.851781	3.338	1.851868	3.453
801	400	1.855112	2.025	1.851660	2.815	1.851686	3.014

Table 5.4 presents the convergence rate results for different timestepping schemes for small volatility ($\sigma = .1$ and $r = .1$). For implicit timestepping linear convergence is recovered ($R = 2$), as expected. However for higher order timestepping schemes such as Crank-Nicolson and second order backward differencing, quadratic convergence is not found (see

Table 5.4).

To try to remedy this problem, the initial payoff function is smoothed out. A classic method for handling discontinuities involves averaging the initial data. Specifically, values at each point are replaced with an average value over nearby space. Mathematically, we set

$$\text{PAYOFF}_{\text{smoothed}}(S_i, A_j) = \int_{K-\frac{\Delta A}{2}}^{K+\frac{\Delta A}{2}} \text{PAYOFF}(S_i, A) dA. \tag{5.43}$$

For a complete description of various smoothing methods the readers are referred to [75].

Table 5.5 Value of a continuously observed fixed strike call Asian call option at the strike with constant timesteps. The initial payoff is smoothed using the average scheme described by equation (5.43). The input parameters are $\sigma = .1$, $r = .1$, $T = .25$, and $K = 100$. Convergence ratios (4.2) are presented for different timestepping schemes: Crank-Nicolson and second order BDF.

		CN timestepping		BDF timestepping	
Size of	No. of	$S = 100$		$S = 100$	
S and A grids	timesteps	Value	R	Value	R
51	25	1.870322	n.a.	1.874276	n.a.
101	50	1.856377	n.a.	1.857462	n.a.
201	100	1.852873	3.981	1.853179	3.925
401	200	1.851963	3.849	1.852053	3.803
801	400	1.851704	3.513	1.851731	3.497

Table 5.5 contains the convergence rate results. From a convergence point of view, the ratios have improved in comparison with the convergence ratio without smoothing (see Table 5.4). However, quadratic convergence is still not obtained. From a theoretical point of view, all the convergence analysis for semi-Lagrangian scheme are based on the smooth properties of the solution [16, 43]. If the solution is smooth then quadratic convergence is recovered. However, if the solution is non-smooth, then we can expect some reduction in the convergence rate.

To confirm our intuition that the non-smooth payoff is in fact the reason why quadratic

Table 5.6 Value of a continuously observed Asian call option at the strike with constant timesteps. The input parameters are $\Delta\tau = .01$, $\sigma = .1$, $r = .1$, $T = .25$ and $K = 100$. Convergence ratios (4.2) are presented for the Crank-Nicolson timestepping scheme.

		Call option (PAYOFF$(A,K) = \mathcal{H}(A-K)(A-K)^2$) $S = 100$	
Size of S and A grids	No. of timesteps	Value	R
51	25	0.081607	n.a.
101	50	0.080608	n.a.
201	100	0.080438	5.827
401	200	0.080408	5.800
801	400	0.080403	5.456

convergence is not recovered, we create an artificial payoff that has the property of being quadratically smooth in the A direction, e.g. PAYOFF$(A,K) = \mathcal{H}(A-K)(A-K)^2$, where $\mathcal{H}(x)$ is the Heaviside function. In this case quadratic convergence is recovered for both Crank-Nicolson and second order backward differencing. Table 5.6 shows detailed convergence results for Crank-Nicolson timestepping.

Several other approaches were considered in an effort to improve convergence. We tried to use Rannacher timestepping [78]; two or more implicit timesteps are taken before reverting to a higher order timestepping scheme such as Crank-Nicolson for example. Numerical experiments indicated that this did not improve the convergence rate. A convergence rate of approximately 3.5 is found in this case. Adaptive timestepping was also considered [44] but this technique did not improve the convergence rate.

5.6.2 American Asian Option under Jump diffusion

In the following we present the convergence results when pricing a continuously observed fixed strike Asian option under the jump diffusion process. We assume that the jumps in the asset price are lognormally distributed [66] (see Chapter 3, equation (3.4)). The input parameters are presented in Table 4.1.

Table 5.7 contains the convergence rate results. We observe that quadratic convergence is still not obtained with the added diffusion due to the integral term.

Table 5.7 Value of a continuously observed fixed strike Asian call option at the strike under the jump diffusion process, constant timesteps. The initial payoff is smoothed using the average scheme described by equation (5.43). The input parameters are presented in Table 4.1. Convergence ratios (4.2) are presented for different timestepping schemes: implicit, Crank-Nicolson and second order BDF.

		CN timestepping		BDF timestepping	
Size of	No. of	$S = 100$		$S = 100$	
S and A grids	timesteps	Value	R	Value	R
51	25	2.417380	n.a.	2.420687	n.a.
101	50	2.406342	n.a.	2.407350	n.a.
201	100	2.403495	3.877	2.403782	3.738
401	200	2.402749	3.816	2.402834	3.761
801	400	2.402539	3.554	2.402564	3.521

Table 5.8 Value of a continuously observed fixed strike put American Asian option (under jump diffusion) with constant timestepping. Crank-Nicolson timestepping is used. The input parameters are defined in Table 4.1. This table presents convergence rates with and without jumps. Iterations refers to the total (over all timesteps) of the maximum number of iterations required for any value of j (see Algorithm 3.3) at each timestep. For the no jump case the Asian American put option is priced using $\sigma = \sigma_{\text{imp}} = .1886$.

		No jump			Jumps		
Size of	No. of	No. of	$(S = 100)$		No. of	$(S = 100)$	
S and A grids	timesteps	iterations	Value	R	iterations	Value	R
51	25	77	2.220443	n.a.	99	2.044636	n.a.
101	50	160	2.195726	n.a.	167	2.018530	n.a
201	100	319	2.188555	3.447	340	2.012220	4.138
401	200	692	2.186717	3.903	716	2.010691	4.126
801	400	1397	2.186243	3.874	1609	2.010281	3.728

Table 5.8 compares the value of an American Asian fixed strike put option with the value of an American Asian fixed strike put option when the underlying stock follows the jump diffusion process described by [66]. Second order backward timestepping is used and the

initial payoff is smoothed out using equation (5.43). We observe that quadratic convergence is not recovered. The convergence ratios are ≈ 3.5.

5.7 Summary

In this chapter we have put forward several contributions.

- We have demonstrated that a semi-Lagrangian method can be used to price continuously observed Asian options. The implementation suggested here reduces this problem to solving a decoupled set of one dimensional discrete partial differential equations (PIDEs) at each timestep. This makes implementation of this method very straightforward in a software library which is capable of pricing discretely observed path dependent options [95].

- We have shown that in the fully implicit case, the semi-Lagrangian method is algebraically identical to a standard numerical method for pricing discretely observed Asian options, when the observation interval is equal to the discrete timestep. Since lattice methods [89] can be regarded as explicit finite difference methods it follows that the usual binomial forest method for Asian options [53] can also be regarded as an explicit semi-Lagrangian method.

- Since the discretized problem at each timestep reduces to a set of decoupled 1d PIDEs, we can make use of the techniques developed in Chapter 3 and in [44, 29, 32] to prove certain properties of the discrete scheme, including convergence of the iterative method used to solve the implicit discrete equations. In the fully implicit case, it is straightforward to prove l_∞ stability.

- We have included experimental computations which indicate that, even if second order timestepping methods are used, observed convergence as the mesh and timestep

is refined occurs at a sub-second order rate for small values of volatility. The problem can be traced to the non-smoothness of the payoff function.

It is straightforward to extend the semi-Lagrangian approach of this chapter to more exotic Asian options such as American Asian options under the jump diffusion process. For a detailed study of these particular contracts, readers are referred to [30].

Chapter 6

A Real Options Approach to Network Management: An Application to Wireless Networks

This chapter applies modern financial option valuation methods to the problem of wireless network capacity investment decision timing. In particular, given a cluster of base stations (with a certain traffic capacity per base station), we determine when it is optimal to increase capacity for each of the base stations contained in the cluster. We express this in terms of the fraction of total cluster capacity in use, i.e. we calculate the optimal time to upgrade in terms of the ratio of observed usage to existing capacity. We study the optimal decision problem of adding new capacity in the presence of stochastic capacity usage. Based on the real options formulation of Chapter 2, a five factor algorithm that captures many of the constraints of wireless network management is developed. This optimal upgrade algorithm will maximize the value of the investment to the network operator.

6.1 Introduction

In Chapter 2 (see Section 2.3) a real options approach to value investment into telecommunication infrastructure was presented. Given the stochastic process described by the following equation

$$\begin{aligned} dQ &= \alpha(\eta - Q)dt + (J-1)QdY \\ d\eta &= \eta\mu dt + \eta\sigma dZ, \end{aligned} \tag{6.1}$$

the value of an investment V depending on Q, η and time τ is given by

$$V_\tau = \alpha(\eta - Q)V_Q + R + \lambda\int_0^\infty V(\eta, QJ, \tau)g(J)dJ + \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta} + \eta(\mu - \zeta\sigma)V_\eta - (r+\lambda)V. \tag{6.2}$$

The derivation of equation (6.2) is given in Chapter 2, Section 2.3. Solving equation (6.2) presents several challenges; it is a two dimensional PDE with no diffusion in the capacity usage direction Q and it contains an integral term. In Chapter 3, we presented numerical techniques for handling the integral term. In Chapter 5, a semi-Lagrangian approach was introduced to solve a two dimensional PDE (6.2) with no diffusion in one space like direction. As a result, the numerical techniques developed in those chapters can now be applied to the wireless network capacity investment decision problem.

Given a cluster of base stations (see Figure 6.1), our ultimate objective is to determine the percentage (in terms of the ratio of observed usage to existing capacity) at which it is optimal to add capacity to each of the base stations in the cluster. This optimal upgrade decision will maximize the value of the investment to the network operator. Figure 6.1

provides a representation of a simplified cluster of base stations. A typical cluster contains at least 20 base stations. When there is too much traffic in a cluster, customers experience blocking (i.e. calls do not get through). For example, during the power outage of August 2003, wireless clusters were under extremely high traffic. As a result many calls were not completed.

A simplistic solution to blocking would be to conduct a traffic study at the base station level and increase the capacity of the stations that are experiencing too much blocking. However, due to the Code Division Multiplexing Access (CDMA) [57] technology that is currently used in leading edge wireless networks, it is not possible to only add capacity to the base stations where high blocking occurs. A user on a network using CDMA technology may talk simultaneously to many base stations since a procedure called *soft hand-off* is used [57]. As such, when there is too much blocking on a particular base station, all the base stations within the cluster must have their capacity increased to maintain homogeneity.

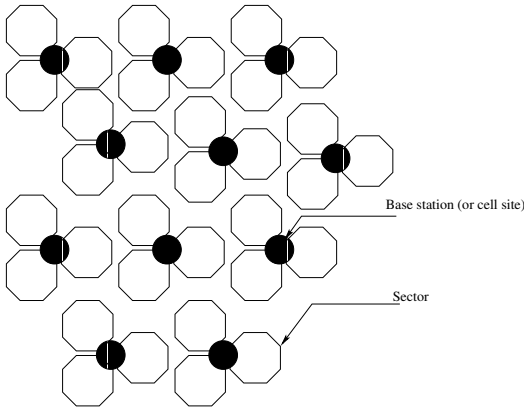


FIGURE 6.1: *Example of a cluster. A typical cluster contains at least 20 base stations. The coverage area or sector is generally not octagonal, but this diagram gives an idea of how a cluster look like.*

With this in mind, we can characterize both traffic and grade of service/blocking probability. Traffic is measured in units of *Erlangs* [61]. An Erlang is defined as the average number of simultaneous calls, or equivalently, the total usage during a time interval divided by the length of that interval. Most network management systems measure usage during

a one hour interval. The *blocking probability* is the probability that a call is blocked when there is no channel available. The blocking probability is evaluated for the load during bouncing busy hours. In practice, network managers study the load during bouncing busy hours and then decide whether or not new capacity must be deployed. Based on the Erlang mathematical models (Erlang-B or Erlang-C [61]), the relationship between blocking probability and capacity usage can be established. In this work we are going to use a blocking probability of 2% [64].

Initially, we assume that we have an existing cluster where each base station has a base level of capacity (i.e. capacity level zero). Using Erlang tables, this means 13.2 Erlang of traffic can be handled at 2% blocking at each base station of the cluster. If capacity is added to a base station (i.e. level one base station), 40 users can be accommodated, meaning 31 Erlang of traffic can be handled at 2% blocking. The base station load at 2% blocking more than doubles when capacity is added. Table 6.1 gives a synopsis of an Erlang table. Consequently, when considering the cluster of base stations, the capacity of the cluster is simply the capacity of a base station multiplied by the number of base stations in the cluster.

Table 6.1 Correspondence table between increment levels at the base station, users, Erlang and minutes of traffic in the hour that can be approximately handled by a base station at 2% blocking [64].

level	Users	Erlang	Minutes of traffic in the hour
0	20	13.2	2376
1	40	31.0	5580
2	60	49.6	8928

6.2 Mathematical Model

This section explores different discretization methods for the PIDE (6.2) using the semi-Lagrangian approach presented in Chapter 5. We use an unequally spaced grid for the PDE discretization in η coordinates $[\eta_0, \ldots, \eta_{imax}]$, and similarly an unequally spaced grid is used in the Q direction $[Q_0, \ldots Q_{jmax}]$. Let

$$V_{i,j}^n = V(\eta_i, Q_j, \tau^n) \tag{6.3}$$

denote the solution with mean η_i, capacity usage Q_j, and time level n. Let $\mathcal{G}$ be the differential operator represented by

$$\mathcal{G}V \equiv \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta} + \eta(\mu - \zeta\sigma)V_\eta - (r+\lambda)V + R, \tag{6.4}$$

and define $\mathcal{I}V$ as

$$\mathcal{I}V \equiv \lambda \int_0^\infty V(\eta, QJ)g(J)dJ. \tag{6.5}$$

Equation (6.2) can then be rewritten as

$$V_\tau - \alpha(\eta - Q)V_Q = \mathcal{G}V + \mathcal{I}V. \tag{6.6}$$

Let us denote the discrete form of $\mathcal{G}V$ or $(\mathcal{I}V)$ at $\eta = \eta_i, Q = Q_j, \tau = \tau^n$ by $(\mathcal{G}V)_{i,j}^n$ or $(\mathcal{I}V)_{i,j}^n$. The Lagrangian derivative along a trajectory $Q = Q(\eta, \tau)$ (η =const) is

$$\frac{DV}{D\tau} = \frac{\partial V}{\partial \tau} + \frac{\partial V}{\partial Q}\frac{dQ}{d\tau}. \tag{6.7}$$

Along the trajectory

$$\frac{dQ}{d\tau} = \alpha(Q - \eta), \tag{6.8}$$

equation (6.2) can be written as

$$\frac{DV}{D\tau} = \mathcal{G}V + \mathcal{I}V. \tag{6.9}$$

Let $Q(\eta_i, Q_j, \tau^{n+1}, \tau)$ be a trajectory satisfying equation (6.8), which passes through the discrete grid point (η_i, Q_j) at $\tau = \tau^{n+1}$. Let $V^n_{i,j(i,n+1)} = V(\eta_i, Q^n_{j(i,n+1)}, \tau^n)$ be the value of the investment at the departure point of the trajectory $Q(\eta_i, Q_j, \tau^{n+1}, \tau)$, at $\tau = \tau^n$. Note that $Q^n_{j(i,n+1)}$ will not necessarily coincide with a grid point Q_j. $Q^n_{j(i,n+1)}$ is determined by solving

$$\frac{dQ}{d\tau} = \alpha(Q - \eta_i), \quad Q^n_{j(i,n+1)} = Q^{n+1}_j + \int_{\tau^{n+1}}^{\tau^n} \left(\frac{dQ}{d\tau}\right) d\tau \tag{6.10}$$

from $\tau = \tau^{n+1}$ to $\tau = \tau^n$. In general, the integration in equation (6.10) must be computed numerically and the integration method should have an error one order higher than the timestepping method used to approximate the Lagrangian derivative (see [2] and Chapter 5).

The characteristic curve associated with equation (6.8) is defined by

$$\frac{dQ}{d\tau} = \alpha(Q - \eta_i), \tag{6.11}$$

and consequently,

$$Q = \eta_i + C\exp(\alpha\tau), \tag{6.12}$$

where C is constant. At $\tau = \tau^{n+1}$, $Q = Q^{n+1}_j$, so that

$$\begin{aligned} \text{At time } \tau^n \quad &: \quad Q^n_{j(i,n+1)} = \eta_i + (Q^{n+1}_j - \eta_i)\exp\left(\alpha(\tau^n - \tau^{n+1})\right), \\ \text{At time } \tau^{n-1} \quad &: \quad Q^{n-1}_{j(i,n+1)} = \eta_i + (Q^{n+1}_j - \eta_i)\exp\left(\alpha(\tau^{n-1} - \tau^{n+1})\right), \end{aligned} \tag{6.13}$$

where $T \geq \tau^{n+1} > \tau^n > \tau^{n-1}$. Discretization of equation (6.9) along the characteristic for

different timestepping schemes gives,

- *fully implicit timestepping*

$$\frac{V_{i,j}^{n+1} - V_{i,j(i,n+1)}^{n}}{\Delta\tau} = (\mathcal{G}V)_{i,j}^{n+1} + (\mathcal{I}V)_{i,j}^{n+1}, \tag{6.14}$$

- *Crank-Nicolson timestepping*

$$\frac{V_{i,j}^{n+1} - V_{i,j(i,n+1)}^{n}}{\Delta\tau} = \frac{1}{2}\left[(\mathcal{G}V)_{i,j}^{n+1} + (\mathcal{I}V)_{i,j}^{n+1}\right] + \frac{1}{2}\left[(\mathcal{G}V)_{i,j(i,n+1)}^{n} + (\mathcal{I}V)_{i,j(i,n+1)}^{n}\right], \tag{6.15}$$

- *second order backward differencing (BDF)* (constant timestepping)

$$\frac{\frac{1}{2}V_{i,j}^{n+1} - 2V_{i,j(i,n+1)}^{n} + \frac{1}{2}V_{i,j(i,n+1)}^{n-1}}{\Delta\tau} = (\mathcal{G}V)_{i,j}^{n+1} + (\mathcal{I}V)_{i,j}^{n+1}. \tag{6.16}$$

Let us define the matrix G such that

$$\left[G \cdot V_j^{n+1}\right]_i = (\mathcal{G}V)_{i,j}^{n+1} + \text{ truncation error }, \tag{6.17}$$

where V_j^{n+1} is the vector of discrete solution values $\left[V_j^{n+1}\right]_i = V(\eta_i, Q_j, \tau^{n+1})$ for fixed Q_j. Using finite differences, the matrix (6.17) is given by

$$(G \cdot V^{n+1})_i = -(\alpha_i + \beta_i + r + \lambda)V_i^{n+1} + \beta_i V_{i+1}^{n+1} + \alpha_i V_{i-1}^{n+1} + R_i^{n+1} \; ; i = 1, \ldots, imax - 1, \tag{6.18}$$

where α_i, β_i are chosen such that the type of approximations used for the derivatives and second derivatives (e.g. central, forward or backward differencing) guarantees that α_i and β_i are non-negative. For $i = 0$, the matrix (6.18) reduces to a diagonal entry with $\alpha_0, \beta_0 = 0$, and for the row $i = imax$, we set $\alpha_i = \beta_i = 0$, and set $\lambda = 0$, and we impose a Dirichlet

boundary condition (see Chapter 3, Section 3.3 for details of α_i and β_i).

As discussed in Chapter 3, the integral term $(\mathcal{I}V)_{i,j}$ can be efficiently computed using interpolation and an FFT. Effectively, $(\mathcal{I}V)$ is approximated by

$$\begin{aligned}(\mathcal{I}V)_{i,j}^{n+1} &\simeq \lambda \sum_k h_{ik} V_{kj}^{n+1} + \text{ truncation error} \\ &= \lambda [H \cdot V_j^{n+1}]_i + \text{ truncation error} \\ & \qquad 0 \le h_{ik} \le 1 \text{ and } \sum_k h_{ik} \le 1. \end{aligned} \tag{6.19}$$

Note that the jumps occur along the demand direction Q; this contrasts with the model of Chapter 5, where jumps occur along the asset direction S.

Let Φ^{n+1} be the Lagrange interpolation operator such that

$$(\Phi^{n+1} \cdot V^n)_{i,j} = V(\eta_i, Q^n_{j(i,n+1)}, \tau^n) + \text{ interpolation error.} \tag{6.20}$$

In the following, for simplicity we describe the method used to solve the discrete equations (6.14), (6.15) and (6.16) only for the fully implicit (6.14) timestepping method. The readers should have no difficulty generalizing the results to the Crank-Nicolson (6.15) and the BDF (6.16) timestepping cases.

In the fully implicit case, the matrix form of the discrete equation (6.14) is given by

$$\left[I - \Delta\tau G - \lambda \Delta\tau H\right] V_j^{n+1} = \left[\Phi^{n+1} V^n\right]_j . \tag{6.21}$$

In order to solve equation (6.21), an iteration scheme (see Algorithm 6.1) is used at each timestep to avoid solving the dense system defined by $\left[H \cdot V_j^{n+1}\right]_i$. Each iteration of Algorithm 6.1 requires a tridiagonal factor and solve, and a forward and back FFT for each j.

Algorithm 6.1 Fixed point iteration scheme to evaluate the jump integral term implicitly.

for $k = 0, 1, 2, \ldots$ until convergence **do**
  // Compute the integral term using FFT along the Q_j direction
  // for each node η_i
  Let $(V_j^{n+1})^0 = (V_j^n)$, $j = 0, 1, \ldots, jmax$
  for $j = 0, 1, \ldots, jmax$ **do**
    Let $\hat{V}_j^{\,k} = (V_j^{n+1})^k$
    Solve
    $[I - \Delta\tau G]\,\hat{V}_j^{\,k+1} = \left[\Phi^{n+1}V^n\right]_j + \lambda\Delta\tau H\hat{V}_j^{\,k}$
  end for
  if $\max_{i,j} \frac{(|\hat{V}_{i,j}^{k+1} - \hat{V}_{i,j}^{k}|)}{\max(1, |\hat{V}_{i,j}^{k+1}|)} < tolerance$ **then**
    Quit
  end if
end for

Theorem 6.1 (Stability of fully implicit discretization). *Assuming that matrix* $[I - \Delta\tau G - \lambda\Delta\tau H]$ *is an* M*-matrix,* Φ *is a linear interpolant, and Dirichlet boundary conditions are used at* $Q = Q_{jmax}$*, then the discretization* (6.14) *is unconditionally stable.*

Proof. This proof is omitted, since it is virtually identical to the proof of Theorem 3.1. □

Theorem 6.2 (Convergence of the fixed point iteration (Algorithm 6.1)). *Assuming that matrix* $[I - \Delta\tau G]$ *is an* M*-matrix (this follows from Lemma D.3 in Appendix D), that* H *has properties* (6.19)*, and Dirichlet boundary conditions are used at* $Q = Q_{jmax}$*, then Algorithm 6.1 is globally convergent to the unique solution of equation* (6.21)*.*

Proof. The proof is virtually identical to the proof of Theorem 3.3. □

Remark 6.1. *Theorem 6.2 can be easily extended to the Crank-Nicolson and BDF cases. For Crank-Nicolson timestepping, we can prove algebraic stability as in Theorem 3.2.*

6.3 Algorithm

Our ultimate objective is to determine the optimal time for an equipment upgrade. This is found as a byproduct of calculating the value of the network operator's investment in equipment. We use a dynamic programming approach to maximize the value of this investment, and this in turn will give us the optimal upgrade strategy.

For ease of exposition we begin by ignoring the upgrade decision entirely and simply describe how to calculate the value of an investment for a fixed level of capacity. Let $\overline{Q}$ be the capacity level. We consider an investment horizon of $T = 5$ years [64]. At this date, the value of the investment is assumed to be given by

$$V(\eta, Q, \overline{Q}, T) = f(\eta, Q).$$

Our methods can be used with any suitable choice of $f(\eta, Q)$, but for simplicity we assume that the value of all capital investment at T is equal to its salvage value, and we further assume that this salvage value is zero (i.e. $f(\eta, Q) = 0$). Implicitly, we are assuming that new technology renders all existing equipment obsolete at T. Our dynamic programming approach solves backwards from this investment horizon date to today. We suppose that revenues R accrue continuously, and are given by $R(\eta, Q, \overline{Q}, \tau)$, measured in dollars per year. Equation (6.2) can be written as

$$V_\tau = \alpha(\eta - Q)V_Q + R(\eta, Q, \overline{Q}, \tau) + \lambda \int_0^\infty V(\eta, QJ, \tau)g(J)dJ + \frac{1}{2}\eta^2\sigma^2 V_{\eta\eta} + \eta(\mu - \zeta\sigma)V_\eta - (r + \lambda)V. \quad (6.22)$$

Equation (6.22) is the valuation equation for a given capacity level $\overline{Q}$, where the network operator receives continuous revenues. The next step is to introduce maintenance costs. We assume that these are constant over time and are paid at discrete time intervals τ_{maint} (e.g.

monthly). With time running backwards, let τ^-_{maint} and τ^+_{maint} denote $\tau^-_{\text{maint}} = \tau_{\text{maint}} - \epsilon$ and $\tau^+_{\text{maint}} = \tau_{\text{maint}} + \epsilon$ where $\epsilon > 0$ and $\epsilon << 1$. Equation (6.22) is then augmented by the condition

$$V(\eta, Q, \overline{Q}, \tau^+_{\text{maint}}) = V(\eta, Q, \overline{Q}, \tau^-_{\text{maint}}) - M\Delta\tau_{\text{maint}}, \tag{6.23}$$

where M is the maintenance cost in dollars per year and $\Delta\tau_{\text{maint}}$ is the time interval between maintenance payments (expressed as a fraction of a year).

Finally, the possibility of upgrading to a higher level of capacity must be taken into account. This adds considerable complexity. We need to keep track of the maximum capacity of each cluster. This will be a discrete variable $\overline{Q}_m$, where there are $m_{\max}$ possible capacity levels, indexed by $m = 0, 1, \ldots, m_{\max} - 1$. We allow upgrades to any higher level of capacity (e.g. we could upgrade to the highest possible level, skipping all intermediate levels [31]). In other words, we could upgrade from $\overline{Q}_m$ to $\overline{Q}_u$, where $u \in \{m+1, \ldots, m_{\max} - 1\}$.

The decision of whether or not to upgrade (and to which level) is assumed to be made periodically, at a discrete set of observation times $t_{\text{obs}} = \{0, \Delta t_{\text{obs}}, 2\Delta t_{\text{obs}}, \ldots\}$. Typically, we will let $\Delta t_{\text{obs}} = 1$ month. If a decision is made to upgrade at some time $t_{\text{up}} \in t_{\text{obs}}$, then the actual upgrade is completed at $t_{\text{up}} + \vartheta$. We assume that $\vartheta / \Delta t_{\text{obs}}$ is an integer. Note that ϑ is the amount of time necessary to order and set up the equipment.

Due to the fact that the dynamic programming approach works backwards in time, at any time in t_{obs} we cannot know when (or if) an upgrade decision was made, and to which capacity level. Hence, we have to solve for all possible times at which an upgrade could occur, and all possible capacity levels for an upgrade. In addition to the variable u described above which indicates the capacity level of a potential upgrade, we need an additional discrete counter variable l to track the times at which upgrades might occur. l ranges from $0, 1, \ldots, l_{\max}$ where $l_{\max} = \frac{\vartheta}{\Delta t_{\text{obs}}}$. $l = 0$ corresponds to the value of the investment at the existing level of capacity. The value of $l > 0$ corresponds to the amount of time elapsed

(working backwards and measured in terms of $\Delta\tau_{\text{obs}} = \Delta t_{\text{obs}}$) since a potential upgrade was completed. For example, suppose that $\vartheta = 3$ months ($\Delta\tau_{\text{obs}} = 1$ month), and $l_{\max} = 3$. If $l = 1$, then the equipment came on-line 1 month ago, and the upgrade decision will be made 2 months from now.[1] Similarly, if $l = 2$, the equipment was on-line 2 months ago, but the actual decision of whether or not to upgrade will occur 1 month from now. If $l = 3$, the decision to upgrade will be made immediately, and the equipment was operational 3 months ago. Note that at each observation time $\tau_{\text{obs}} = T - t_{\text{obs}}$, the value of l changes. Let τ^- (τ^+) denote backwards time right before (after) an observation time, and let l^- and l^+ be the value of the counter variable at τ^- and τ^+. Then we have

$$l^+ = l^- + 1 \text{ if } l^- < l_{\max}. \tag{6.24}$$

This is because the elapsed time will be incremented by one if we have not yet reached the time when the upgrade decision occurs.

Our valuation function now depends on six variables: the mean reversion level η, the level of capacity usage Q, the existing capacity level $\overline{Q}_m$, the upgrade decision indicator u, (which indicates that a decision has been made to upgrade to capacity $\overline{Q}_u$) the discrete time counter variable l, and backwards time τ: $V = V(\eta, Q, \overline{Q}_m, u, l, \tau)$ (as well as the periodic maintenance costs). We use the notation $u = m, l = 0$ to indicate the value of the investment at the existing level of capacity.

The only remaining factor to include is the cost of upgrading. Let $C_{m\to u}(\tau)_{l\to l-1}$ denote some (possibly time-dependent) designated partial payment from l to $l-1$ of the total cost of upgrading from a cluster with maximum capacity $\overline{Q}_m$ to one with maximum capacity

[1] Recall again that time is running backwards in the dynamic programming approach being described here. In terms of forward time, the upgrade decision has been made in the past and the equipment will be installed in the future; but in backwards time the upgrade decision will be made in the future and the equipment installation has occurred in the past.

$\overline{Q}_u$.[2] We assume that these upgrade costs are paid at the observation times τ_{obs}, implying that

$$V(\eta, Q, \overline{Q}_m, u, l, \tau_{\text{obs}}^+) = V(\eta, Q, \overline{Q}_m, u, l-1, \tau_{\text{obs}}^-) - C_{m\to u}(\tau^-)_{l\to l-1}, \quad l = 2, \ldots, l_{\max} \tag{6.25}$$

where $u = m+1, \ldots, m_{\max}-1$, $\tau_{\text{obs}}^+ = \tau_{\text{obs}} + \epsilon$, and $\tau_{\text{obs}}^- = \tau_{\text{obs}} - \epsilon$, $\epsilon > 0$, and $\epsilon << 1$. At $l = 1$, we have

$$V(\eta, Q, \overline{Q}_m, u, 1, \tau_{\text{obs}}^+) = V(\eta, Q, \overline{Q}_u, u, 0, \tau_{\text{obs}}^-). \tag{6.26}$$

Note that we have assumed that upgrade costs are prepaid (going forward in time) so that $C_{m\to u}(\tau^-)_{1\to 0} = 0$. Equations (6.26) and (6.25) indicate that partial upgrade costs are assumed to be paid at the beginning of the month (going forward in time) and it is possible to upgrade from $\overline{Q}_m$ to $\overline{Q}_u$, but the new equipment will not be ready for some time. For example, suppose that $\vartheta = 3$ months, $\Delta\tau_{\text{obs}} = 1$ month and $l_{\max} = 3$, then upgrade costs are paid at times $0, 1, 2$ going forward in time (see Figure 6.2). Thus going backward in time equation (6.25) represents partial payments for positions $2, 1$ (i.e. $l = 3, 2$) and equation (6.26) indicates what occurs at position 3 (i.e. $l = 1$) (see Figure 6.2). Furthermore, as we will see in more detail in the next section, when working backward in time, upgrade costs are paid before upgrade decisions are made, and as a consequence we need to store $V(\eta, Q, \overline{Q}_m, u, l_{\max}, \tau_{\text{obs}}^-)$ into a temporary variable $TMP(\eta, Q, \overline{Q}_m, u, l_{\max}, \tau_{\text{obs}}^-)$ before equation (6.25) is applied. This temporary variable is then used in deciding to upgrade (This will be discussed in detail in Algorithm 6.4).

At each upgrade decision date, τ_{up}, we maximize the value of the investment V by comparing the value of the investment in the current level of capacity with that of all

[2]The designated partial payment can be specified in a variety of ways. For instance, with three months lead time, one-third of the cost could be paid in each of the three months. Alternatively, all of the cost could be paid up front.

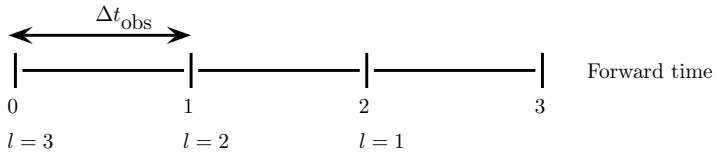


FIGURE 6.2: *This figure illustrates the sequence of payments, suppose that* $\vartheta = 3$ *months,* $\Delta\tau_{obs} = 1$ *month and* $l_{\max} = 3$*, then partial upgrade costs are paid at times* $0, 1, 2$ *going forward in time. Thus going backward in time equation* (6.25) *represents partial payments for* $l = 3, 2$ *and equation* (6.26) *indicates what occurs at time* $l = 1$*.*

possible completed upgrades to higher capacities. This implies

$$V(\eta, Q, \overline{Q}_m, m, 0, \tau_{\text{up}}^+) = \max\left[V(\eta, Q, \overline{Q}_m, m, 0, \tau_{\text{up}}^-), TMP(\eta, Q, \overline{Q}_m, u, l_{\max}, \tau_{\text{up}}^-) - C_{m\to u}(\tau^-)_{0\to l_{\max}}\right], \quad (6.27)$$

for $u = m+1, \ldots, m_{\max} - 1$, $\tau_{\text{up}}^+ = \tau_{\text{up}} + \epsilon$, and $\tau_{\text{up}}^- = \tau_{\text{up}} - \epsilon$, $\epsilon > 0$, and $\epsilon << 1$. Equation (6.27) indicates that a decision to add capacity will only be taken if the value of the investment with the extra capacity exceeds the value of the investment without it. An important assumption implicit in equation (6.27) is that while new equipment is ordered, installed, and tested, the current stream of revenue is not interrupted. In other words, while there is lead time, there is no down time. It would be straightforward to extend the analysis in this thesis to the case of finite down time.

Following the financial option valuation literature, the implication of the extra state variables m, l, and u is that we have to solve a set of partial differential equations of the form of equation (6.22), one for each upgrade possibility. Let $V(\eta, Q, \tau)_{m,u,l}$ denote the value of an investment given the continuous variables η, Q, and the discrete variables m, u,

and l. Then we have to solve a collection of problems

$$\frac{\partial V_{m,u,l}}{\partial \tau} = \alpha(\eta - Q)\frac{\partial V_{m,u,l}}{\partial Q} + R(\eta, Q, \overline{Q}_m, \tau) + \lambda \int_0^\infty V(\eta, QJ, \tau) g(J) dJ + \frac{1}{2}\eta^2\sigma^2 \frac{\partial^2 V_{m,u,l}}{\partial \eta^2} + \eta(\mu - \zeta\sigma)\frac{\partial V_{m,u,l}}{\partial \eta} - (r + \lambda)V_{m,u,l}, \quad (6.28)$$

where the revenue function $R(\cdot)$ now explicitly depends on the capacity level via m, and updating rules analogous to equations (6.23) for maintenance costs[3], (6.25) for upgrade costs, and (6.27) for upgrade decisions are also applied.

6.3.1 Further Details About the Algorithm

In this section, a more complete description of the algorithm that we use is presented. For simplicity, we assume that $\Delta\tau_{\text{obs}} = \Delta\tau_{\text{maint}} = \Delta\tau_{\text{up}}$ so that $\tau_{\text{obs}} = \tau_{\text{maint}} = \tau_{\text{up}}$. Note that $\Delta\tau_{\text{up}}$ corresponds to the time interval between upgrade decisions (e.g. one month, six months, one year). At observation times, then any of the following events can take place:

- maintenance costs are paid;
- partial or complete payments are made for capital expenditures;
- decisions about possible upgrades are made; and
- upgrades come on-line.

In our application, the exact sequence of events at observation times is as follows. Working backwards in time, we assume that maintenance costs are paid first, followed by upgrade costs, partially completed upgrades move one step closer to completion, then upgrade decisions are made, and finally upgrades come on-line. Figure 6.3 graphically presents the order of events that could happen at any observation times.

[3]Note that the maintenance costs are assumed to depend on the capacity level, i.e. $M = M_m$.

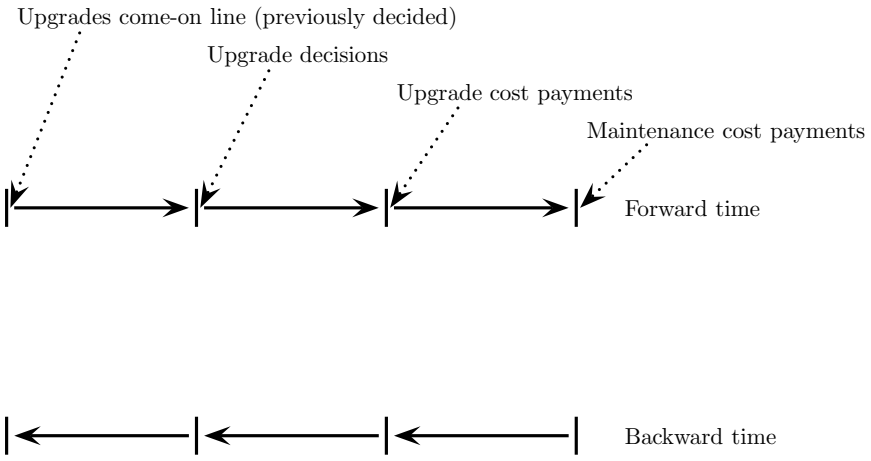


FIGURE 6.3: *Sequence of events at observation times. Working backwards in time, we assume that maintenance costs are paid first, followed by upgrade costs, partially completed upgrades move one step closer to completion, then upgrade decisions are made, and finally upgrades come on-line.*

We now give some more details about the revenue function. As noted above, it is assumed that the owner of the cluster receives continuous payments. For the maximum capacity $\overline{Q}$ (measured in minutes per bouncing busy hour), we have

$$R(\eta, Q, \overline{Q}_m, \tau) = \min(Q, \overline{Q}_m) P(\tau). \tag{6.29}$$

The price function $P(\tau)$ is given by

$$P(\tau) = P_0 \exp\left(-\varphi(T - \tau)\right), \tag{6.30}$$

where $P_0 Q$ has units of dollars per year and φ is a decay parameter. The payment received can be no larger than the maximum capacity of the cluster multiplied by the price. Consistent with our earlier observation (see Figure 2.2 in Chapter 2), we assume that price is a

known decreasing function of time.

Turning to upgrade costs, recall that $C_{m\to u}(\tau)_{l\to l-1}$ is a designated partial payment of the upgrade cost from a cluster of maximum capacity $\overline{Q}_m$ to one with maximum capacity $\overline{Q}_u$. These upgrade costs are assumed to decay exponentially over time with the same parameter β as the price term above.[4]

Next we provide a more complete summary of our algorithm. As noted above, a variety of events can occur at observation times, but for simplicity we denote the time (working backwards) before any of these events as τ^- and the time following the event as τ^+. The length of time between observation times is $\Delta\tau = 1$ month. For illustrative purposes, we assume here that the lead time required for the upgrade is 3 months. We can think of the algorithm as having the following steps:

- **Step 1:** Impose the condition $V_{m,u,l}(\eta, Q, 0) = f(Q)$ at the investment horizon date $\tau = 0$.
- **Step 2:** Solve the collection of PDEs (6.28) to the first observation time.
- **Step 3:** Consider the observation time as a time when maintenance costs are paid. Update the solution by subtracting these costs, as described in the pseudo-code given in Algorithm 6.2.
- **Step 4:** Consider the observation time as partially completed upgrades move one step closer to completion, upgrade costs are paid and completed upgrades to higher capacity levels become available. The solution updating is shown graphically in Figure 6.5. Pseudo-code is given in Algorithm 6.3.
- **Step 5:** Consider the observation time as an upgrade decision date and apply equa-

[4]More precisely, only the total cost of an upgrade is assumed to decline over time. If the upgrade costs are paid at observation dates throughout the lead time interval, then the individual payments for a particular upgrade do not decay exponentially.

tion (6.27). This is sketched in the pseudo-code provided in Algorithm 6.4, and depicted graphically in Figure 6.4.

- **Step 6:** If $\tau = T$, terminate. Otherwise, solve the collection of PDEs to the next observation date and repeat Steps 3-6.

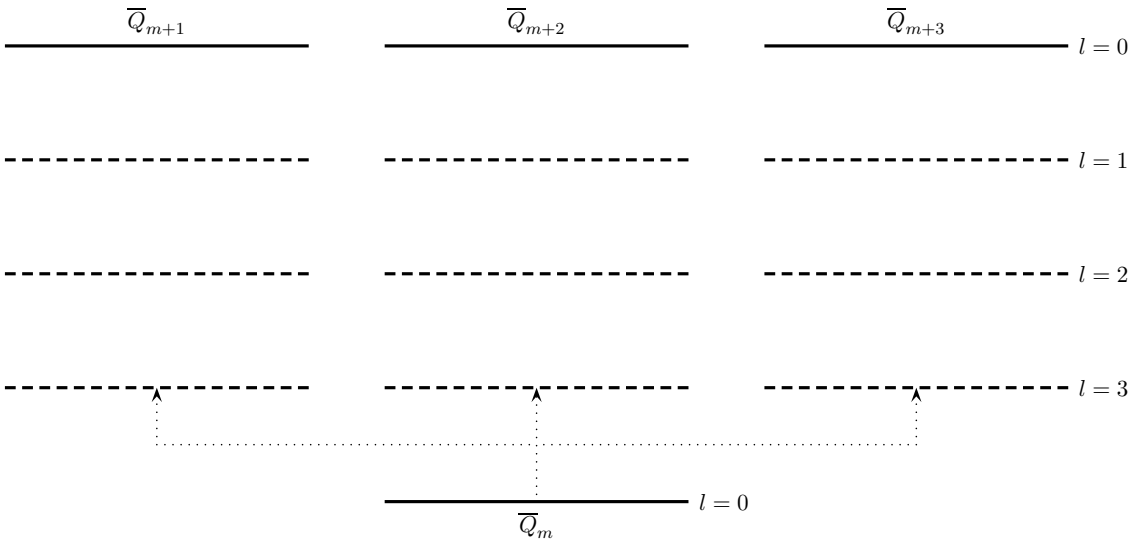


FIGURE 6.4: *Illustration of the upgrade decision. At each upgrade decision date, the value of the investment at the current level of capacity $\overline{Q}_m$ and the current level of capacity usage, $V(\eta, Q, \tau)_{m,m,0}$, is compared with the values of possible completed investments to various higher levels of capacity, $V(\eta, Q, \tau)_{m,u,3}$. In this case, we are considering upgrades to 1, 2, or 3 higher levels (i.e. $u \in m+1, m+2, m+3$), and it takes 3 months to complete the investment. In other words, working backwards in time, we start 3 months before the upgrade decision date with the values of completed investments corresponding to the different higher capacity levels. The counter variable l is incremented at each observation date (and appropriate adjustments are made to the value, arising from items such as capital costs being paid). When 3 months have elapsed, we have reached the upgrade decision date, so we compare the values of the various investments into higher levels of capacity with the current level, and choose the highest value at this time. Note that for ease of exposition η was not indicated in this figure.*

Algorithm 6.2 At each maintenance cost payment date, V is reduced by the amount of these costs.

for $m = 0, \ldots, m_{\max} - 1$ **do**
  $V_{m,m,0}(\eta, Q, \tau^+) = V_{m,m,0}(\eta, Q, \tau^-) - M_m \Delta\tau$
  // loop over the different clusters
  for $u = m + 1, \ldots, m_{\max} - 1$ **do**
    // update the solution with the appropriate maintenance cost
    for $l = 1, \ldots, l_{\max}$ **do**
      $V_{m,u,l}(\eta, Q, \tau^+) = V_{m,u,l}(\eta, Q, \tau^-) - M_m \Delta\tau$
    end for
  end for
end for

Algorithm 6.3 At each observation date, update the solution of the potential upgrades from the cluster with maximum capacity $\overline{Q}_m$ to that with maximum capacity $\overline{Q}_u$.

for $m = 0, \ldots, m_{\max} - 2$ **do**
  // loop over the different clusters
  for $u = m + 1, \ldots, m_{\max} - 1$ **do**
    //copy solution of $l_{\max}$ into a temporary variable
    $TMP_{m,u,l_{max}}(\eta, Q, \tau^-) = V_{m,u,l_{\max}}(\eta, Q, \tau^-)$
    // loop over the upgrade possibilities for cluster m
    for $l = l_{\max}, \ldots, 2$ **do**
      // loop over the elapsed time since the upgrade was completed
      $V_{m,u,l}(\eta, Q, \tau^+) = V_{m,u,l-1}(\eta, Q, \tau^-) - C_{m \to u}(\tau^-)_{l \to l-1}$
    end for
    //at l=1 (upgrade cost are prepaid in forward time)
    $V_{m,u,1}(\eta, Q, \tau^+) = V_{u,u,0}(\eta, Q, \tau^-)$
  end for
end for

6.4 Parameter Estimation

This section describes the estimation of parameter values for the no-jump case. Details regarding estimates of the growth rate μ and the volatility σ can be found in Chapter 2, Table 2.3. The volatility σ and the growth rate μ for the minutes per bouncing busy hour have been estimated by averaging σ and μ for the different time series presented in Table 2.3. Smoothing anomalous jumps in capacity usage, it is found that $\mu \approx .30$ per year and $\sigma \approx .65$ per year$^{\frac{1}{2}}$.

Algorithm 6.4 At each upgrade decision date, compare the value of the investment if a decision is taken to upgrade to a higher level of capacity with the value of the existing level of capacity.

for $m = 0, \ldots, m_{\max} - 2$ **do**
  // loop over the cluster with capacity $\overline{Q}_m$
  for $u = m+1, \ldots, m_{\max} - 1$ **do**
    // loop over the upgrade possibilities for cluster m
    $V_{m,m,0}(\eta, Q, \tau^+) = \max\left[V_{m,m,0}(\eta, Q, \tau^-), TMP_{m,u,l_{\max}}(\eta, Q, \tau^-) - C_{m\to u}(\tau^-)_{0\to l_{\max}}\right]$
  end for
end for

6.4.1 Revenue

We assume that the owner of the cluster receives continuous payments. The data underlying Figure 2.2 (see Chapter 2) indicates that the average revenue per user (ARPU) is approximatively .229$/min. However, this value corresponds to the total revenue received based on daily traffic, and not net revenue based on bouncing busy hour traffic. Consequently, to estimate P_0 we need to adjust the average revenue per user (ARPU) appropriately. The total revenue producing minutes in a day is approximately ten times the bouncing busy hour minutes [64] and net revenue per user represents approximately seventy percent of the average revenue per user: $.229 \times .7 \approx .1603$ [64]. P_0 is given by

$$\begin{aligned} P_0 = {} & \text{ARPU} \times 10 \text{ (busy hr factor adjustment [64])} \\ & \times .7 \text{ (marginal revenue)} \times 250 \text{ (weekends not revenue producing)}, \end{aligned}$$

where P_0 has units $\left[\frac{\text{dollars}}{\frac{\text{minutes}}{\text{bbhr}}\text{year}}\right]$. The maximum total revenue per year for various increment levels for a cluster is given in Table 6.2.

6.4.2 Market price of risk

We next consider the market price of risk ζ. In Appendix G (G.1, G.2), two approaches are presented to estimate the market price of risk. The market price of risk is estimated assum-

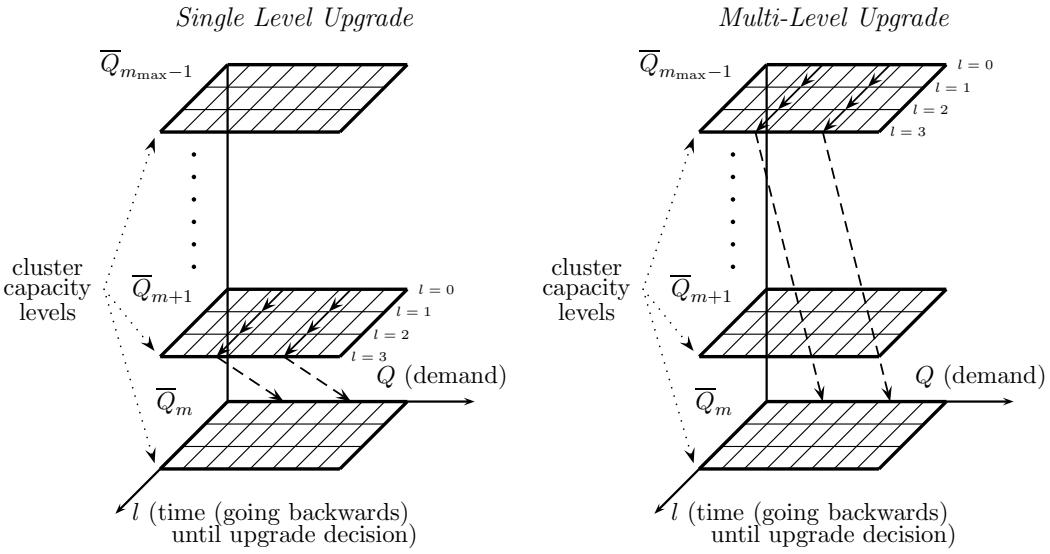


FIGURE 6.5: *Flow of information at observation dates. In the left panel, a possible upgrade decision from level $\overline{Q}_m$ to the next higher level, $\overline{Q}_{m+1}$ is being evaluated. In the right panel, the potential upgrade is to the highest possible level, $\overline{Q}_{m_{\max}-1}$. Time is running in the backwards direction, and it is assumed that it takes three months for a capacity upgrade to be implemented once the decision has been taken. Consider, for example, the left panel. The value of a completed investment into the higher capacity level $\overline{Q}_{m+1}$ is shown along the line along the Q-axis (i.e the $l = 0$ line) in the $\overline{Q}_{m+1}$ plane. At the next monthly observation date, the value is moved one step in to the $l = 1$ line. Note that on this line, the capacity in use is actually $\overline{Q}_m$, because the upgrade is incomplete. Similar updating occurs at the succeeding observation dates, until the $l = 3$ line is reached. At this point, the decision of whether or not to upgrade to this capacity level is made by comparing the value of the investment along this line with that along the Q-axis for the existing $\overline{Q}_m$ capacity level. If a decision is made to upgrade to capacity level $\overline{Q}_{m+1}$, the value from the $\overline{Q}_{m+1}$ level along the $l = 3$ line migrates down to the $l = 0$ line in the $\overline{Q}_m$ plane, as indicated by the dashed line. The evaluation of the possible multi-level upgrade in the right panel is similar. Note that for ease of exposition η was not indicated in the Figure.*

Table 6.2 Maximum total revenue per year for a cluster based on bouncing busy hour traffic.

Cluster increment level	Revenue for a cluster in $ per year
0	19,043,640
1	44,723,700
2	71,557,920

ing that traffic demand Q follows a geometric Brownian motion stochastic process. In the case where we ignore jumps in demand to determine today's upgrade decision percentage, the methods presented in Appendix G are reasonable. In contrast, when considering jumps in demand it is recognized that the methods used to estimate the market price of risk may no longer be appropriate. However, given the number of data points available, we think that these methods (Appendix G) are enough to provide a rough estimate of the market price of risk.

Using these two approaches, we find that ζ lies between .03 and .17, $\zeta \approx .1$ corresponding to the average value. This interval of values is due to several factors such as tax rate assumptions, and available number of data points; details can be found in Appendix G. While these estimates may be seen as quite different, it turns out that our results (in terms of when it is optimal to upgrade capacity) are not too sensitive to $\zeta \in \{.03, .10, .17\}$.

6.4.3 Upgrade and Maintenance Costs

The hardware cost of a single capacity increment to a base station is approximately $\$150,000$. Consequently adding capacity for each base station of the cluster would cost $\$150,000 \times 20 = \$3,000,000$.

Once an order has been placed to upgrade a cluster with additional capacity, it takes approximately two months before the hardware is delivered, one month to install the hardware, and one month to set up and optimize the equipment. Hence, it takes about four months from order placement until the equipment is on-line.

The monthly maintenance fee of a cluster is the cost of a base station technician and a T1 cable connection. A technician maintains approximately 20 base stations. The cost of a base station technician is assumed to $150,000 per year. Base station leasing costs are approximately $1,500 per month and T1 back-haul costs are approximately $500 per month per T1, and there are three T1 connections per base station. Electricity and warranty costs

are respectively estimated as \$250 and \$200 per base station. Thus the total maintenance cost for the cluster cost per year is about \$150,000 + \$1,500 × 20 × 12 + \$500 × 20 × 12 + (\$250 + \$200) × 20 × 12 = \$738,000. Table 6.3 contains a summary of the maintenance costs.

Table 6.3 Maintenance costs for a cluster in dollars per year. The maintenance costs vary depending on capacity installed on the cluster.

Cluster increment level	Annual cluster maintenance cost
0	\$738,000
1	\$858,000
2	\$978,000

For voice traffic, the equipment price is decreasing every year, but not by a significant amount. Vendors offer features to increase the traffic handling capability of each base station every couple of years. By offering enhancements, the vendor feels justified in keeping the dollars per new equipment per base station rate relatively stable. Consequently, we assume that the upgrade costs decrease by the same percentage as the spot price (i.e. 5% per year) [64]. Table 6.4 presents the maximum number of minutes of traffic that can be handled for different cluster capacities.

Table 6.4 Maximum number of minutes of traffic in the hour that can be handled at 2% blocking for the cluster (20 base stations).

Increment level	One base station	Cluster (20 base stations)
0	2376	47520
1	5580	111600
2	8928	178560

6.5 Results

We now provide some illustrative results. Before proceeding any further, the following two observations should be made. First, we are concentrating on the upgrade decision, not the value of the investment. Our algorithm determines both of these to maximize the value of the investment, hence we can determine the optimal upgrade decision. This is equivalent to identifying the location of a free boundary. This boundary will move over time. In fact, it will generally increase as time moves forward, as shown in the bandwidth capacity management context evaluated in [31]. The intuition for this is straightforward. As we get closer to the investment horizon T, there is less time available for the higher revenues associated with higher traffic and expanded capacity to cover the costs of upgrading. Of course, things are more complicated due to factors such as upgrade costs decreasing over time and the possibility of upgrading to more than a single higher capacity level, but this is the basic idea. Note, however, that if we get close enough to T (within the lead time required to install the extra capacity), we will never upgrade, no matter how high demand is, simply because the new capacity will never provide any revenue to offset the expansion costs. Keeping this in mind, we will present results only for the upgrade decision at $t = 0$.[5] Second, we will not concentrate here on the effects of the parameters of the traffic time series, μ and σ. The effects of these two parameters are qualitatively similar to those reported in [31] in the bandwidth context. Faster rates of the growth parameter μ obviously give rise to faster upgrades to capture the extra revenue earlier. Higher levels of the volatility parameter σ work in the opposite direction, with increased uncertainty, it is better to wait longer before upgrading. This is because the probability of traffic demand decreasing substantially is higher with higher volatility.

[5] In fact, the optimal upgrade decision as a percentage of existing capacity in use does not change too much for $0 \leq t \leq 2$, provided we are considering fairly frequent evaluations of the upgrade decision. If we suppose that upgrade decisions are made only on annual basis, then we find more significant differences.

Since we are solving the set of partial differential equations (6.28) on a discrete grid of points and over a discrete set of timesteps, we must study the convergence rate of our different numerical schemes (i.e. implicit, Crank-Nicolson or BDF). The convergence ratio R is defined in the following way. For each test, as we double the number of grid points in both the η and Q directions, we cut the timesteps ($\Delta\tau$) in half. Let $\Delta\tau = \max(\tau^{n+1} - \tau^n)$, $\Delta Q_{\max} = \max(Q_{j+1} - Q_j)$. If we then carry out a convergence study, letting $h \to 0$, where $\Delta\eta_{\max} = \text{Const. } h$, $\Delta Q_{\max} = \text{Const. } h$, and $\Delta\tau = \text{Const. } h$. then we can assume that the error in the solution (at a given node) is

$$V_{\text{approx}}(h) = V_{\text{exact}} + \text{Const. } h^{\xi}.$$

The convergence ratio is then defined as

$$R = \frac{V_{\text{approx}}(h/2) - V_{\text{approx}}(h)}{V_{\text{approx}}(h/4) - V_{\text{approx}}(h/2)}. \tag{6.31}$$

In the case of quadratic convergence ($\xi = 2$), then $R = 4$, while for linear convergence ($\xi = 1$), $R = 2$.

6.5.1 No Jump Case

Table 6.5 Model parameters.

Investment horizon (T)	5 years
Decay in price (β)	.05/year
Growth rate (μ)	.3/year
Volatility (σ)	.65/year$^{\frac{1}{2}}$
Risk free rate (r)	.04/year
Market price of risk (ζ)	[.03; .10; .17]/year
Speed of reversion (α)	10^6/year

We begin by considering the case where jumps in traffic Q are ignored, and thus $\alpha = \infty$ and $\lambda = 0$. (In the actual computation, the parameter α (i.e speed of reversion) is set to 10^6 (effectively infinity)). In effect we are assuming that demand Q is following the geometric Brownian motion process of mean η (6.1) and ignoring jumps ($\lambda = 0$). Table 6.5 presents a summary of the estimated parameters. We assume that all the cluster capacity (at 2% blocking) is available and the stream of revenues is capped when the maximum capacity is reached.

It was found in Chapter 5 that for continuously observed fixed strike Asian options, the initial discontinuity in the initial condition and the absence of convection led to suboptimal quadratic convergence. However, when solving the investment timing problem, the initial conditions are smooth and as a consequence quadratic convergence should be recovered. Table 6.6 presents our convergence results for different timestepping schemes. As expected linear convergence is found for implicit timestepping (i.e. $R = 2$) and quadratic convergence is obtained for both Crank-Nicolson and second order backward differencing (i.e. $R = 4$).

Table 6.6 Investment value at $\eta = Q = 100 = \frac{\overline{Q}}{2}$ when considering upgrading a cluster of maximum capacity $\overline{Q} = 47520$ and $\zeta = .03$. Constant timestepping is used and the timestep $dt = 5./250$ (i.e. one week). At each grid refinement, the timestep is halved. The input parameters are provided in Table 6.5. The convergence ratio R is defined in equation (6.31). The equipment lead time is four months and upgrade decisions are taken monthly.

	Implicit timestepping		CN timestepping		BDF timestepping	
Size of η and Q grids	Value ($\times 10^7$)	R	Value ($\times 10^7$)	R	Value ($\times 10^7$)	R
71	5.3130	n.a.	5.3037	n.a.	5.3034	n.a.
141	5.3045	n.a.	5.2852	n.a.	5.2849	n.a.
281	5.3000	1.908	5.2792	3.048	5.2788	3.055
561	5.2984	2.760	5.2775	3.657	5.2772	3.670

We also studied the effect of the grid size on today's optimal upgrade percentage. It is found that for grids larger than 281 points, today's upgrade percentage does not change by

more than one percent. Consequently, in the following, the effect of the upgrade decision frequency on today's upgrade decision is studied with a grid containing 281 points in both the capacity usage Q and mean η direction.

Table 6.7 Today's upgrade decision in terms of upgrade percentage with respect to the maximum capacity $\overline{Q} = 47520$ of the cluster at 2% blocking. We allow 100% usage of the total cluster capacity. Above 100%, the revenue stream is capped by the cluster maximum capacity. The input parameters are provided in Table 6.5. It takes four months between the time the equipment is ordered and the time it is on-line. 281 points are used for both the η and Q grids. The results are accurate to within plus or minus one percent.

Upgrade decision	One increment level			Two increment level		
interval	$\zeta = .03$	$\zeta = .1$	$\zeta = .17$	$\zeta = .03$	$\zeta = .1$	$\zeta = .17$
monthly	90%	94.75%	99.5%	101.25%	106.75%	112.5 %
quarterly	77.5%	82 %	86.5%	87.5%	92.25%	97.5%
semi-annually	72.5%	76.25%	81%	81%	86%	91%
annually	60%	65%	70%	67.5%	72.5%	77.5%

Table 6.7 contains results for three different values of the market price of risk ζ, for a variety of upgrade decision intervals, ranging from monthly to annually. We notice that for the higher value of $\zeta = .17$, the percentage, in terms of the maximal cluster capacity at which it is optimal to upgrade, is higher than when $\zeta = .03$. This result is in accordance with our modeling framework, since as ζ increases, the drift term $(\mu - \zeta\sigma)$ of equation (6.22) decreases, and the upgrade should occur later. In Table 6.7, we observe that the difference between $\zeta = .03$ and the average value of the market price of risk $\zeta = .1$, in terms of the optimal upgrade percentage, is approximately 5%. Similar results are found when comparing $\zeta = .17$ and $\zeta = .1$. This indicates that our upgrade timing results are not too sensitive to our estimate of ζ, however the value of the investment on the other hand is quite sensitive to ζ.

In Table 6.7, we find that today it would be optimal to add one level of capacity to each

base station of the cluster if $90-99\%$ of its maximum capacity is reached when considering monthly upgrade decision dates. As the upgrade decision interval is changed from monthly upgrade decisions to annual upgrade decisions, the upgrade percentage decreases from 90% to 60%. Intuitively, this simply reflects the fact that with less frequent decision opportunities it is better to upgrade earlier. This behavior is consistent with the results found in [26, 33, 31]. In Table 6.7, we observe that in some cases it is optimal to wait before upgrading the cluster until traffic demand is above its maximum capacity (e.g. monthly upgrade, $\kappa = .17, > 100\%$). While these levels may appear somewhat surprising and unrealistic, we are in effect simply saying that customers must be experiencing a lot of blocking before it is optimal to upgrade the cluster. In practice, since there is a financial penalty associated with poor quality of service, this level of blocking would probably not occur.

As a final comment about Table 6.7, we observe that in some circumstances it can be optimal to add two new carriers per cell site, rather than one. This is the case if traffic demand is quite high, and particularly if upgrade decisions are made relatively infrequently.

6.5.2 Jump Case

We now consider the case when jumps in capacity usage are no longer ignored. Before proceeding any further, jumps in capacity usage must be characterized. We assume jumps are log normally distributed (this prevents negative demand and this is a common assumption in financial applications [66])

$$g(J) = \frac{\exp\left(-\frac{(\log(J)-\nu)^2}{2\gamma^2}\right)}{\sqrt{2\pi}\gamma J}, \tag{6.32}$$

where ν corresponds to the mean and γ to the standard deviation.

In Appendix H, an empirical method to estimate the jump frequency mean and standard deviation is derived. This method assumes that the demand Q follows a geometric Brownian

motion stochastic process and the jumps corresponds to extreme events. However, since only large deviations from the mean are considered, the standard deviation for the traffic jumps is certainly biased upward. It is recognized that the approach used here may not be complete, other approaches exist (see [1, 6, 38, 40, 49, 51] and references therein). However given the number of data points available, we believe that our procedure is sufficient to assess the effects of jumps in traffic on the optimal wireless investment decision problem

Table 6.8 presents our parameter estimates. Recall that the time series considered represents week to week capacity usage. To prevent intra-week usage patterns, the day of the week with highest average capacity usage was selected and that same day was chosen for the rest of the time series. As such, when estimating jump frequency only one day per week was considered. Consequently, to adjust the value of the jump frequency, we assume that jumps are uniformly distributed for each day of the week and thus multiply the jump frequency obtained in Appendix H by five (bouncing busy hour traffic is only monitored during business days). Admittedly, this is a very crude estimation. With this assumption, it is found that there are approximately 28 jumps in traffic per year, with mean $\nu \approx -.0508$/year and standard deviation $\gamma \approx .2372$/year.

Table 6.8 Jump parameters summary table for different time series (representing different switches in Toronto). This table shows the estimates of jumps which occur on only one business day (Thursdays). To estimate the jump frequency on any business day, we multiply the Thursday jump frequency by five.

Time series	# iterations	Jump frequency (Thursday) per year	Jump mean ν	Jump standard deviation γ
A	1	5	-0.1284	0.3961
B	3	6	-0.0121	0.2304
C	4	6	-0.0119	0.0850
Average	n.a.	≈ 5.6	$\approx$ -.0508	≈ 0.2372

Note that for the time series B and C, jumps in traffic that were previously considered as part of the "normal" capacity usage have been discovered. As a consequence, the volatility

Table 6.9 Drift rate and volatility summary table for different time series (different switches in Toronto).

Time series	Drift rate μ %/(year)	Volatility σ per %/(year)$^{\frac{1}{2}}$
A	-24.75	90.69
B	21.72	41.91
C	69.35	21.22
Average	$\approx$ 22.11	$\approx$ 51.27

σ and drift rate μ have to be adjusted. Table 6.9 presents these new values which can be compared with the drift rate and volatility numbers reported in Table 2.3. The speed of reversion α is estimated by inspecting the different time series A, B and C. It is empirically observed that when there is a jump in capacity usage Q, traffic usage returns to the vicinity of its pre-jump values the next day. Thus, α is estimated to be ≈ 250. Table 6.10 presents a summary of the input parameters when considering jumps in traffic.

Table 6.10 Model parameters.

Investment horizon (T)	5 years
Decay in price (β)	.05/year
Growth rate (μ)	.2211/year
Volatility (σ)	.5127/year$^{\frac{1}{2}}$
Risk free rate (r)	.04/year
Market price of risk (ζ)	[.03;.10;.17]/year
Speed of reversion (α)	250/year
Jump frequency (λ)	28/year
Jump mean (ν)	-.0508/year
Jump standard deviation (γ)	.02372/year

Since we are solving a set of partial differential equations on a discrete grid of points and over a discrete set of timesteps, as in the previous section convergence testing must be carried out. Linear convergence is found for implicit timestepping (i.e. $R = 2$) and quadratic convergence is obtained for both Crank-Nicolson and second order backward differencing

(i.e. $R = 4$, tables are omitted).

Table 6.11 Today's upgrade decision in terms of upgrade percentage with respect to the maximum capacity $\overline{Q} = 47520$ of the cluster at 2% blocking. We allow 100% usage of the total cluster capacity. Above 100%, the revenue stream is capped by the cluster maximum capacity. The input parameters are provided in Table 6.10. It takes four months between the time the equipment is ordered and the time it is on-line. 281 points are used for both the η and Q grids. These results are accurate to within plus or minus one percent.

Upgrade decision	One increment level			Two increment level		
interval	$\zeta = .03$	$\zeta = .1$	$\zeta = .17$	$\zeta = .03$	$\zeta = .1$	$\zeta = .17$
monthly	95.75%	99.75%	103.75%	106.5%	111.%	116.%
quarterly	85.5%	89.%	93%	95%	99.5%	104%
semi-annually	81%	84%	88%	90%	94%	98.75%
annually	70%	73.75%	77.5%	77.5%	82%	87%

Next, we study the influence of jumps on today's upgrade decision when considering different upgrade decision intervals and market prices of risk. Comparing Table 6.11 with Table 6.7, we observe that adding jumps in capacity usage delays today's upgrade decision by approximately 7%. These results are not surprising since both the drift rate and volatility have changed and, furthermore, jumps in capacity usage have been included. These jumps are on average negative in size. This implies that jumps in capacity usage do have a significant effect on today's upgrade decision (see Tables 6.11 and Table 6.7).

However, a concern which could be raised regarding this analysis is that the assumption of 100% allowed capacity usage (at 2% blocking) is quite aggressive. However, our framework allows us to take into account criteria such as quality of service. For instance, it is conceivable that engineers prefer a safety buffer between the maximum capacity (at 2% blocking) and the capacity available to customers. Once this threshold is reached, the quality of service deteriorates. To compensate, customers may receive rebates or free calls. Of course, customers may also seek other vendors. These effects are explored in the next

section.

6.5.3 Quality of Service Modeling

We adopt the following simple model to investigate how quality of service can affect the upgrade decision. Let ϕ be the safety factor, representing a percentage of the maximum capacity of the cluster. Then revenue is specified as

$$R(\eta, Q, \overline{Q}_m, \tau) = \begin{cases} R(\eta, Q, \overline{Q}_m, \tau)\,(= Q \times P(\tau)) & \text{if } Q \leq \phi\overline{Q_j} \\ P(\tau)\overline{Q_m}\max(1 - \frac{Q-\phi\overline{Q_m}}{\phi\overline{Q_m}}, 0.0) & \text{otherwise.} \end{cases} \tag{6.33}$$

For example, $\phi = .9$ implies that only 90% of the maximum capacity of any given cluster (at 2% blocking) can be used. Note that the above expression implies that revenues eventually drop to zero as capacity usage keeps increasing. Effectively, we are adding a financial penalty as the quality of service deteriorates. It might also be possible to develop penalty functions based on the effect of quality of service on customer "churn rates" (i.e. the loss of customers to other vendors as a result of poor service), or other criteria. This differs from the cases considered above in Table 6.7 where revenue was simply capped once capacity was reached.

In Table 6.12, we present the results for several simulations with the safety level ranging from 100% to 80% when the upgrade decision is considered every six months for $\zeta = .1$. The table considers the case where we add a single carrier to each cell site. Similar results are obtained when upgrading involves adding two carriers per cell site. The first row in the table, where the safety factor is indicated as "none", corresponds to the situation in the previous tables where revenue is simply capped when demand is higher than available capacity.

We observe that as the safety level decreases, the upgrade occurs sooner in terms of the

Table 6.12 Today's upgrade decision with different safety factor (6.33) values in terms of upgrade percentage with respect to the maximum capacity of the cluster at 2% blocking. Input parameters are provided in Table 6.5 and the market price of risk is given by $\zeta = .1$. The equipment lead time is four months. 281 points are used for both η and Q grids. Jumps in traffic are assumed to follow a lognormal distribution [65]. Upgrade decisions are taken every six months.

	One cluster level	
Safety factor	$(\lambda = 0, \alpha = 10^6, \sigma = .65, \mu = .3)$	$(\lambda = 28, \alpha = 250, \sigma = .5127, \mu = .2211)$
None	76.25%	84%
100%	65%	72.5%
90%	62.5%	70%
80%	60%	67.5%

percentage of the total capacity of the cluster. Similar behavior is reported when upgrade decisions are made for shorter time intervals (e.g. quarterly, or monthly). This phenomenon is intuitively correct: if there is a financial penalty for poor quality of service, upgrades will occur sooner. Once again we find that adding jumps does affect today's upgrade decision . When comparing the columns of Table 6.12, we note that jumps in capacity usage tend to delay the decision to upgrade.

6.6 Summary

In this chapter several contributions are put forward.

- We considered the issue of management of a wireless network under uncertain capacity usage. While the method presented here is similar to the one described in [31], the limitations embedded in the algorithm in that work are alleviated by developing a five factor algorithm. This enables us to consider different upgrade decision intervals independent of equipment lead time.

- We showed that today's upgrade decision, in terms of the optimal upgrade percentage, is affected by the value of the market price of risk. As the value of the market price of risk increases, today's upgrade decision percentage increases effectively delaying the decision to upgrade.

- We studied the effect of jumps in capacity usage on today's upgrade decision. It is found that jumps in capacity usage delays today's upgrade decision by approximately 7%.

- We have demonstrated that by developing appropriate penalty functions which assign a cost to poor quality of service, we can combine both financial and quality of service criteria. This approach requires managers to assign a cost to quality of service issues. Penalty functions could be real financial incentives provided to users (e.g. during high blocking periods, all calls are free), or they could be based on customer churn rates.

Chapter 7

Conclusion

In this thesis, a model for investment in wireless network capacity was developed. Based on a real options approach, a two dimensional partial integro-differential equation (PIDE) was derived to determine the value of an investment into a wireless network under uncertain capacity usage. This PIDE presented several numerical challenges. In particular, the integral term of the PIDE had to be solved accurately and quickly enough such that the general PIDE solution was reasonably accurate. The absence of diffusion in the capacity usage direction of the two dimensional PIDE also had to be handled carefully to avoid stability and accuracy problems. To validate our numerical algorithms, problems from the financial derivative pricing literature were used.

To address the numerical problems due to the integral term of the PIDE, we considered the one dimensional pricing PIDE used when pricing American options under the jump diffusion process. We developed an iterative method for solving the discrete penalized equations. We derived sufficient conditions for the global convergence of this iteration (at each timestep). Unlike previous work, the method developed here used implicit timestepping for both the correlation integral term and the American constraint. As a result we obtained high order convergence (in terms of timestepping error). A sufficient condition for global

convergence of the iterative method for solving the discretized penalized jump diffusion equations (at each timestep) was developed.

To address the numerical problems due to the absence of diffusion in the two dimensional real options PIDE, we considered the two dimensional PIDE for pricing continuously observed fixed strike Asian options under jump diffusion. We demonstrated that a semi-Lagrangian method could be used effectively. The implementation suggested reduced this problem to solving a decoupled set of one dimensional discrete partial integral differential equations (PIDEs) at each timestep. We showed that in the fully implicit case, the semi-Lagrangian method was algebraically identical to a standard numerical method for pricing discretely observed Asian options, when the observation interval was equal to the discrete timestep. Furthermore, since the discretized problem at each timestep reduced to a set of decoupled one dimensional PIDEs, we proved convergence of the iterative method used to solve the implicit discrete equations.

A five factor algorithm was developed to determine the optimal timing of capacity investment in a wireless network. We computed the optimal time to upgrade in terms of the ratio of observed usage to existing capacity. This optimal upgrade algorithm maximized the value of the investment to the network operator. We showed that today's upgrade decision, in terms of the optimal upgrade percentage, increases as the value of the market price of risk increases. We studied the effect of jumps in capacity usage on today's upgrade decision, it was found that jumps in capacity usage delayed today's upgrade decision. We also demonstrated that by developing appropriate penalty functions which assigned a cost to poor quality of service, we could combine both financial and quality of service criteria. This approach requires managers to assign a cost to quality of service issues. Penalty functions could be real financial incentives provided to users (e.g. during high blocking periods, all calls are free), or they could be based on customer churn rates.

Although, we were initially motivated by the PIDE results from the real options model

of wireless network investments. The numerical algorithms developed in this thesis have immediate application in finance. Specifically, we have developed robust numerical methods for pricing continuously observed path dependent options, where the underlying follows a jump diffusion process.

7.1 Future Work

There are several possible avenues for future research.

- A detailed analysis of pricing and hedging various types of exotic options under a jump diffusion process could be carried out.

- It would be interesting to extend the analysis to more complex models for the evolution of the underlying state variable. Among the candidates here are more general Lévy processes than the jump diffusion case, or multifactor models such as those recently explored in [42], which feature stochastic volatility with Poisson jumps in both the state variable itself and its volatility.

- It would be desirable to prove convergence of iterative schemes without the use of M-matrices. For single factor options, the commonly used $V_{SS} = 0$ boundary condition destroys the M-matrix property. For two factor options (such as stochastic volatility models), the M-matrix property no longer holds if there is a non-zero correlation between the asset price and the volatility [93]. However, we have observed that the penalty method for imposing the American constraint appears to be globally (and rapidly) convergent for models with stochastic volatility, but no jumps [93].

- Finally, the real options modelling framework could be extended to include uncertain parameters, along the same lines as used for uncertain volatility [74].

Appendix A

Jump Diffusion PIDE Derivation

Let S represent the underlying stock price. Movements in this variable over time are assumed to be described by a jump diffusion process of the form

$$\frac{dS}{S} = \mu dt + \sigma dZ + (J-1)dY, \tag{A.1}$$

where μ is the drift rate, σ is the volatility associated with the continuous (Brownian) component of the process, dZ is the increment of a Gauss-Wiener process, dY is a Poisson process. Let $V(S,t)$ be the value of a contingent claim that depends on the underlying stock price S (A.1) and time t.

Let Π be a portfolio of derivatives in the presence of jumps, containing the option V and $-\Delta$ of the underlying stock:

$$\Pi = V - \Delta S. \tag{A.2}$$

Using the extended version of Ito's lemma [89], the change in value of the portfolio defined by equation (A.2) is given by

$$d\Pi = dV - \Delta dS, \tag{A.3}$$

or

$$d\Pi = \left(V_t + \frac{1}{2}\sigma^2 S^2 V_{SS}\right) dt + (V_s - \Delta)\, dS + (V(SJ,t) - V(S,t) - S\Delta(J-1))\, dY. \quad \text{(A.4)}$$

Equation (A.4) contains two sources of uncertainty. The first source of uncertainty is due to the Wiener part of the stock dZ, while the second source of uncertainty is due to the Poisson process dY. Setting $\Delta = V_s$ eliminates the diffusive risk, and equation (A.5) becomes

$$d\Pi = \left(V_t + \frac{1}{2}\sigma^2 S^2 V_{SS}\right) dt + (V(SJ,t) - V(S,t) - S\Delta(J-1))\, dY. \quad \text{(A.5)}$$

The portfolio Π evolves deterministically except for the Poisson part. In [66] it is argued that if the jumps in the asset price (A.1) are uncorrelated with the market as a whole, then the risk of jumps should not be priced into the options. Diversifiable risk should not be rewarded. Consequently, taking the expectation equation (A.5) becomes

$$d\Pi = \left(V_t + \frac{1}{2}\sigma^2 S^2 V_{SS}\right) dt + E\left[V(SJ,t) - V(S,t)\right] \lambda dt - V_s S E[(J-1)]\lambda dt, \quad \text{(A.6)}$$

where $E[.]$ is the expectation operator such that

$$E[x] = \int x g(x) dx,$$

where $g(x)$ a probability density function. Consequently during the time interval dt, the portfolio Π of derivatives must grow at the risk free rate $d\Pi = r\Pi dt$. Equation (A.6) becomes

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa) S V_S + \lambda E\left[V(SJ,t) - V(S,t)\right] - rV, \quad \text{(A.7)}$$

where $\kappa = E[J-1]$. For future convenience, note that equation (A.7) can be rewritten in

a slightly different form as

$$V_t + \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r + \lambda)V + \lambda \int_0^\infty V(SJ, t)g(J)dJ = 0. \tag{A.8}$$

Merton's assumption about jump risk being diversifiable does not hold for index based options, and in this case one must use an equilibrium based method [60] or a mean variance hedging approach [9]. The partial integro differential equations resulting in either case are essentially identical. In this case, the parameters λ, and κ are risk adjusted not historical [5].

Appendix B

Error Estimates for Correlation Integral

In this Appendix, we show how to extend the domain of integration of the integral (3.67) such that FFT wrap-around effects are less than a user specified tolerance. To avoid algebraic complication, we derive the results in an informal way. We focus on the error due to the FFT wrap-around. We assume that any other errors (interpolation, discretization of the integral, etc.) are second order in the asset grid spacing, and we ignore such errors in the following.

We make the assumptions

$$\begin{aligned} y_{\min} &< 0 \\ y_{\max} &> 0 \\ \overline{V}(y) &\geq 0 \\ \overline{V}(y) &\leq \max(A_2, A_1 e^y) \\ \overline{f}(y) &\leq A_3 e^{-\gamma|y|}, \quad \forall y, \quad \gamma > 2 \\ \max \overline{V}(y) &\leq A_4 \ ; \ y \in [y_{min}, y_{max}] \end{aligned} \tag{B.1}$$

where A_1, A_2, A_3, A_4 are constants independent of y. We assume that $\overline{V}(y)$ is only given at discrete points on the interval $y \in [y_{\min}, y_{\max}]$.

Recall that we wish to compute an approximation to

$$I(x) = \int_{y_{\min}}^{y_{\max}} \overline{V}(x+y)\overline{f}(y)dy. \tag{B.2}$$

Considering the case where $x = y_{\max}$, equation (B.2) becomes

$$\begin{aligned} I(y_{\max}) &= \int_{y_{\min}}^{y_{\max}} \overline{V}(y_{\max}+y)\overline{f}(y)\ dy \\ &= \int_{y_{\min}}^{0} \overline{V}(y_{\max}+y)\overline{f}(y)\ dy + \int_{0}^{y_{\max}} \overline{V}(y_{\max}+y)\overline{f}(y)\ dy. \end{aligned} \tag{B.3}$$

When using an FFT to evaluate the correlation integral, the term

$$\int_{0}^{y_{\max}} \overline{V}(y_{\max}+y)\overline{f}(y)\ dy$$

is actually evaluated using

$$\int_{0}^{y_{\max}} \overline{V}(y_{\min}+y)\overline{f}(y)\ dy \tag{B.4}$$

due to the wrap-around effect of the discrete FFT. The idea here is to extend the definition of $\overline{V}$ to the interval $y \in [y_{\min} - \Delta y^-, y_{\max} + \Delta y^+]$. We make the assumption that the values of $\overline{V}$ can be obtained in the extended regions correct to second order, and we ignore these errors in the following. Setting $\Delta y^- = 0$ for the time being, equation (B.3) becomes

$$I(y_{\max}) = \int_{y_{\min}}^{\Delta y^+} \overline{V}(y_{\max} + y)\overline{f}(y)\ dy + \int_{\Delta y^+}^{y_{\max}} \overline{V}(y_{\max} + y)\overline{f}(y)\ dy. \tag{B.5}$$

Now, the wrap-around error $E(y_{\max})$ which will occur using an FFT will be

$$\begin{aligned}
E(y_{\max}) &\simeq \int_{\Delta y^+}^{y_{\max}} |\overline{V}(y_{\max} + y) - \overline{V}(y_{\min} + \{y - \Delta y^+\})|\overline{f}(y)\ dy \\
&\leq \max\left[\int_{\Delta y^+}^{y_{\max}} A_1 e^{(y_{\max}+y)}\overline{f}(y)\ dy\ ,\ A_2 \int_{\Delta y^+}^{y_{\max}} \overline{f}(y)\ dy\right] \\
&\leq \max\left[A_1 e^{y_{\max}} \int_{\Delta y^+}^{y_{\max}} e^y A_3 e^{-\gamma y}\ dy\ ,\ A_2 \int_{\Delta y^+}^{y_{\max}} A_3 e^{-\gamma y}\ dy\right] \\
&\leq \max\left[A_1 e^{y_{\max}} \frac{e^{\Delta y^+} A_3 e^{-\gamma \Delta y^+}}{\gamma - 1}\ ,\ A_2 A_3 \frac{e^{-\gamma \Delta y^+}}{\gamma}\right] \\
&\leq A_3 e^{\Delta y^+} e^{-\gamma \Delta y^+} \max\left[A_1 e^{y_{\max}}, A_2\right] \\
&\leq A_3 e^{-\gamma \Delta y^+} e^{\Delta y^+} A_4
\end{aligned} \tag{B.6}$$

So, if we require that the relative error at $x = y_{\max}$ be less than a given tolerance, then we select Δy^+ such that

$$\frac{E(y_{\max})}{A_4} \leq A_3 e^{-\gamma \Delta y^+} e^{\Delta y^+} < tol_R. \tag{B.7}$$

For practical purposes, we assume that $\overline{f}(\Delta y^+) \simeq A_3 e^{-\gamma \Delta y^+}$, so that we can approximate equation (B.7) by

$$\frac{E(y_{\max})}{A_4} \quad \simeq \quad \overline{f}(\Delta y^+) e^{\Delta y^+} < tol_R. \tag{B.8}$$

Note that a relative error criteria is a reasonable choice at $x = y_{\max}$ since $\overline{V}(y_{\max})$ may be

$O(e^{y_{\max}})$.

Following the same reasoning at $x = y_{\min}$, assuming that $\Delta y^+ = 0$ for simplicity, we now extend the domain of $\overline{V}$ to the left by Δy^-, and we assume that we can determine $\overline{V}$ in $[y_{\min} - \Delta y^-, y_{\min}]$ correct to second order. The error in $I(y_{\min})$ due to wrap-around is given by

$$\begin{aligned}
E(y_{\min}) &\simeq \int_{y_{\min}}^{-\Delta y^-} |\overline{V}(y_{\min} + y) - \overline{V}(y_{\max} + y + \Delta y^-)| \overline{f}(y)\ dy \\
&\leq \max \left[\int_{y_{\min}}^{-\Delta y^-} A_1 e^{(y_{max} + y + \Delta y^-)} \overline{f}(y)\ dy\ ,\ A_2 \int_{y_{\min}}^{-\Delta y^-} \overline{f}(y)\ dy \right] \\
&\leq \max \left[A_1 e^{(y_{\max} + \Delta y^-)} \int_{y_{\min}}^{-\Delta y^-} e^y A_3 e^{\gamma y}\ dy\ ,\ A_2 A_3 \frac{e^{-\gamma \Delta y^-}}{\gamma} \right] \\
&\leq \max \left[A_1 e^{(y_{\max} + \Delta y^-)} A_3 e^{-\gamma \Delta y^-} \left(\frac{e^{-\Delta y^-}}{1+\gamma} \right)\ ,\ A_2 A_3 \frac{e^{-\gamma \Delta y^-}}{\gamma} \right] \\
&\leq A_3 e^{-\gamma \Delta y^-} \max \left[A_2, A_1 e^{y_{\max}} \right] \\
&\leq A_3 e^{-\gamma \Delta y^-} A_4
\end{aligned} \tag{B.9}$$

Therefore we can require that the absolute error at $x = y_{\min}$ be less than a specified tolerance if we select Δy^- such that

$$E(y_{\min}) \leq A_4 A_3 e^{-\gamma \Delta y^-} < tol_L. \tag{B.10}$$

Again, for practical purposes we assume that $\overline{f}(-\Delta y^-) \simeq A_3 e^{-\gamma \Delta y^-}$ and so we approximate equation (B.7) to obtain

$$E(y_{\min}) \leq A_4 \overline{f}(-\Delta y^-) < tol_L. \tag{B.11}$$

An estimate of A_4 can be obtained from

$$A_4 \simeq \max_{0 \leq S \leq S_{\max}} V(S, \tau = 0). \tag{B.12}$$

Note that an absolute error criteria is appropriate near $x = y_{\min}$ since $\overline{V}$ is bounded at $y = y_{\min}$.

Typically, we chose $tol_L = tol_R = 10^{-6}$. Since the wrap-around errors are largest at $x = y_{\min}, x = y_{\max}$, selecting the domain extensions which satisfy equations (B.7) and (B.10) will bound these errors at all other points. The domain extensions are illustrated in Figures B.1-B.2.

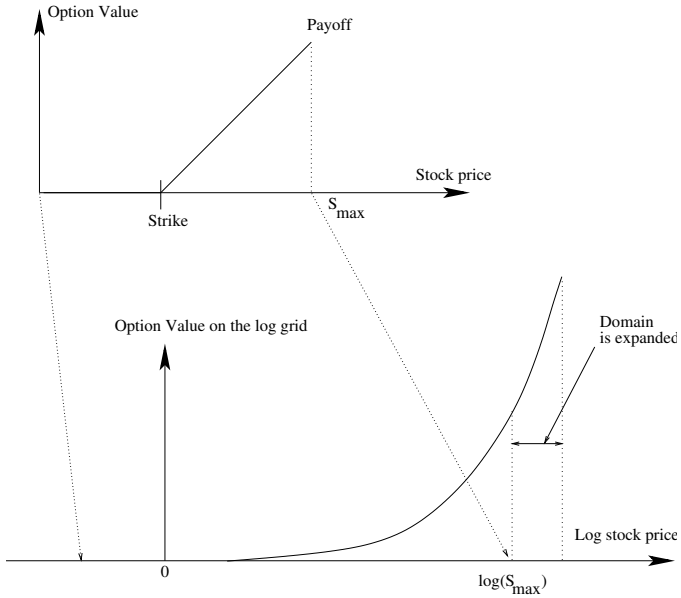


FIGURE B.1: *The value of the option is interpolated onto the log-spaced grid. The right hand side boundary of the log-spaced grid* $y_{\max} = \log(S_{\max})$ *is expanded by* Δy^+*, where* Δy^+ *is given by equation* (B.7).

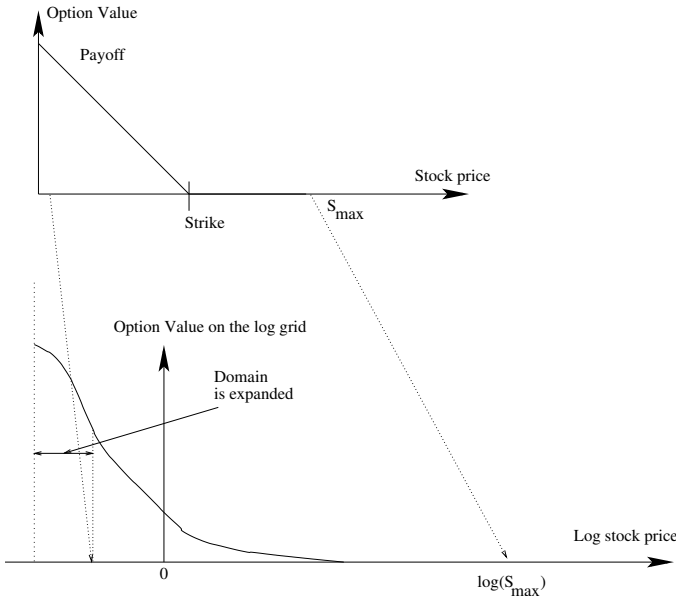


FIGURE B.2: *The value of the option is interpolated onto the log-spaced grid. However, the value of the option* $V(S,\tau)$ *at* $S = 0$ *is not used. The left hand side boundary grid point is chosen to be* $\log S_1$ *where* S_1 *is the grid point nearest to* $S = 0$*. This left boundary is then expanded by* Δy^-*, which is given by equation* (B.10)*.*

Appendix C

Von Neumann Stability Analysis

In this Appendix, we will carry out a Von Neumann stability analysis for Crank-Nicolson timestepping in the special case of constant parameters and an equally spaced grid in $\log S$ coordinates.

From equations (3.5) and (3.13),

$$V_\tau = \frac{1}{2}\sigma^2 S^2 V_{SS} + (r - \lambda\kappa)SV_S - (r+\lambda)V + \lambda \int_{-\infty}^{\infty} \overline{V}(y) f\left(y - \log S\right)\, dy. \tag{C.1}$$

where $\overline{V}(x,\tau) = V(\exp(x),\tau)$ and $\overline{f}(y) = f(\exp(y))$. Using the change of variable $x = \log(S)$ and substituting into (C.1), we obtain

$$\overline{V}_\tau = \frac{1}{2}\sigma^2 \overline{V}_{xx} + (r - \lambda\kappa - \frac{1}{2}\sigma^2)\overline{V}_x - (r+\lambda)\overline{V} + \lambda \int_{-\infty}^{\infty} \overline{V}(y)\overline{f}(y-x)dy. \tag{C.2}$$

From equation (C.2), it can be observed that the integral part of the PIDE is simply a correlation product. Using the correlation operator $\otimes$ from equation (3.14), equation (C.2) can be written as

$$\overline{V}_\tau = \frac{1}{2}\sigma^2 \overline{V}_{xx} + (r - \lambda\kappa - \frac{1}{2}\sigma^2)\overline{V}_x - (r+\lambda)\overline{V} + \lambda\overline{V} \otimes \overline{f}. \tag{C.3}$$

A Crank-Nicolson discretization of equation (C.3) is

$$\frac{\overline{V}_i^{n+1} - \overline{V}_i^n}{\Delta\tau} = \frac{\lambda}{2}\left[(\overline{V}\otimes\overline{f})_i^n + (\overline{V}\otimes\overline{f})_i^{n+1}\right] +$$
$$\frac{1}{2}\left[\frac{1}{2}\sigma^2\left(\frac{\overline{V}_{i+1}^{n+1} - 2\overline{V}_i^{n+1} + \overline{V}_{i-1}^{n+1}}{\Delta x^2}\right) + \left(r - \lambda\kappa - \frac{1}{2}\sigma^2\right)\left(\frac{\overline{V}_{i+1}^{n+1} - \overline{V}_{i-1}^{n+1}}{2\Delta x}\right) - (r+\lambda)\overline{V}_i^{n+1}\right] +$$
$$\frac{1}{2}\left[\frac{1}{2}\sigma^2\left(\frac{\overline{V}_{i+1}^{n} - 2\overline{V}_i^{n} + \overline{V}_{i-1}^{n}}{\Delta x^2}\right) + \left(r - \lambda\kappa - \frac{1}{2}\sigma^2\right)\left(\frac{\overline{V}_{i+1}^{n} - \overline{V}_{i-1}^{n}}{2\Delta x}\right) - (r+\lambda)\overline{V}_i^{n}\right]. \quad \text{(C.4)}$$

Equation (C.4) can be written as

$$\overline{V}_i^{n+1}\left[1 + (\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right] - \frac{\Delta\tau}{2}\beta\overline{V}_{i+1}^{n+1} - \frac{\Delta\tau}{2}\alpha\overline{V}_{i-1}^{n+1}$$
$$= \overline{V}_i^{n}\left[1 - (\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right] + \frac{\Delta\tau}{2}\beta\overline{V}_{i+1}^{n} + \frac{\Delta\tau}{2}\alpha\overline{V}_{i-1}^{n} + \frac{\Delta\tau}{2}\lambda\left[(\overline{V}\otimes\overline{f})_i^n + (\overline{V}\otimes\overline{f})_i^{n+1}\right], \quad \text{(C.5)}$$

where

$$\alpha = \frac{\sigma^2}{2\Delta x^2} - \frac{r - \lambda\kappa - \frac{\sigma^2}{2}}{2\Delta x} \quad \text{(C.6)}$$
$$\beta = \frac{\sigma^2}{2\Delta x^2} + \frac{r - \lambda\kappa - \frac{\sigma^2}{2}}{2\Delta x}. \quad \text{(C.7)}$$

Let $\hat{V}^n = [\overline{V}_0^n, \overline{V}_1^n, \ldots, \overline{V}_p^n]'$ be the discrete solution vector to equation (C.3). Suppose the initial solution vector is perturbed, i.e. $\hat{V}^0 = \overline{V}^0 + E^0$, where $E^n = [E_0^n, \ldots, E_p^n]'$ is the perturbation vector. Note that $E_p^n = 0$ since Dirichlet boundary conditions are imposed at this node. Then, from equation (C.5), we obtain the following equation for the propagation

of the perturbation

$$E_i^{n+1}\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta E_{i+1}^{n+1}-\frac{\Delta\tau}{2}\alpha E_{i-1}^{n+1}$$
$$=E_i^n\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta E_{i+1}^n+\frac{\Delta\tau}{2}\alpha E_{i-1}^n+\frac{\Delta\tau}{2}\lambda\left[(E\otimes\overline{f})_i^n+(E\otimes\overline{f})_i^{n+1}\right]. \tag{C.8}$$

In the following we determine the stability of our discretization scheme using the von Neumann approach [80]. In order to apply the Fourier transform method, we assume that the boundary conditions can be replaced by periodicity conditions. We define the inverse discrete Fourier transform (DFT) as follows (note that we have selected a particular scaling factor)

$$E_i^n=\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}}C_k^n\exp\left(\sqrt{-1}\,\frac{2\pi}{N}ik\right) \tag{C.9}$$

$$f_i=\frac{1}{X_N}\sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}}F_l\exp\left(\sqrt{-1}\,\frac{2\pi}{N}il\right), \tag{C.10}$$

where C_k and F_l correspond respectively to the discrete Fourier coefficients of E and f, and $X_N=x_{N/2}-x_{-N/2+1}$ is the width of the domain along the x-axis. Note that the notation C_k^n should be interpreted as $(C_k)^n$, i.e. in this case n is a power, not a superscript.

The forward transforms are

$$C_k^n=\frac{X_N}{N}\sum_{i=-\frac{N}{2}+1}^{\frac{N}{2}}E_i^n\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}ik\right) \tag{C.11}$$

$$F_l=\frac{X_N}{N}\sum_{i=-\frac{N}{2}+1}^{\frac{N}{2}}f_i\exp\left(-\sqrt{-1}\,\frac{2\pi}{N}il\right). \tag{C.12}$$

The discrete correlation is given by

$$(E \otimes \overline{f})_i^n = \frac{X_N}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} E_j^n f_{j-i}, \tag{C.13}$$

which is second order accurate. Substituting (C.9) and (C.10) into (C.13), we obtain

$$\begin{aligned}(E \otimes \overline{f})_i^n &= \frac{X_N}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jk\right) \frac{1}{X_N} \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}} F_l \exp\left(\sqrt{-1}\,\frac{2\pi}{N}(j-i)l\right) \\ &= \frac{1}{X_N}\frac{1}{N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n F_l \exp\left(-\sqrt{-1}\,\frac{2\pi}{N} il\right) \\ &\times \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jk\right) \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jl\right).\end{aligned}$$

Using the orthogonality condition

$$\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jk\right) \exp\left(\sqrt{-1}\,\frac{2\pi}{N} jl\right) = \begin{cases} N & \text{if } l = -k \\ 0 & \text{otherwise} \end{cases}, \tag{C.14}$$

we find that

$$(E \otimes \overline{f})_i^n = \frac{1}{X_N} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^n F_{-k} \exp\left(\sqrt{-1}\,\frac{2\pi}{N} ik\right). \tag{C.15}$$

Substituting (C.9) and (C.15) into (C.8) gives

$$\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n+1}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]$$
$$-\frac{\Delta\tau}{2}\beta\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n+1}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}(i+1)k\right)-\frac{\Delta\tau}{2}\alpha\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n+1}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}(i-1)k\right)$$
$$=$$
$$\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]$$
$$+\frac{\Delta\tau}{2}\beta\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}(i+1)k\right)+\frac{\Delta\tau}{2}\alpha\frac{1}{X_N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} C_k^{n}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}(i-1)k\right)$$
$$+\frac{\Delta\tau}{2}\lambda\left[\frac{1}{X_N}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}} C_k^{n}F_{-k}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)+\frac{1}{X_N}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}} C_k^{n+1}F_{-k}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)\right] \tag{C.16}$$

Because of linearity, each Fourier component can be treated separately. Equation (C.16)

becomes

$$C_k^{n+1}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)\left[1+(\alpha+\beta+r+\lambda)\,\frac{\Delta\tau}{2}\right]$$
$$-\frac{\Delta\tau}{2}\beta C_k^{n+1}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i+1)k\right)-\frac{\Delta\tau}{2}\alpha C_k^{n+1}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i-1)k\right)$$
$$=$$
$$C_k^{n}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)\left[1-(\alpha+\beta+r+\lambda)\,\frac{\Delta\tau}{2}\right]$$
$$+\frac{\Delta\tau}{2}\beta C_k^{n}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i+1)k\right)+\frac{\Delta\tau}{2}\alpha C_k^{n}\exp\left(\sqrt{-1}\frac{2\pi}{N}(i-1)k\right)$$
$$+\frac{\Delta\tau}{2}\lambda\left[C_k^{n}F_{-k}\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)+C_k^{n+1}F_{-k}\exp\left(\sqrt{-1}\frac{2\pi}{N}ik\right)\right]. \quad \text{(C.17)}$$

Dividing equation (C.17) by $C_k^n\exp\left(\sqrt{-1}\ \frac{2\pi}{N}ik\right)$, we obtain

$$C_k\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta C_k\exp\left(\sqrt{-1}\ \frac{2\pi}{N}k\right)-$$
$$\frac{\Delta\tau}{2}\alpha C_k\exp\left(-\sqrt{-1}\ \frac{2\pi}{N}k\right)-C_k\frac{\lambda\Delta\tau}{2}F_{-k}$$
$$=$$
$$\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta\exp\left(\sqrt{-1}\ \frac{2\pi}{N}k\right)+$$
$$\frac{\Delta\tau}{2}\alpha\exp\left(-\sqrt{-1}\ \frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\lambda F_{-k}. \quad \text{(C.18)}$$

Factoring the C_k term, equation C.18 becomes

$$C_k=\frac{\left[1-(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]+\frac{\Delta\tau}{2}\beta\exp\left(\sqrt{-1}\ \frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\alpha\exp\left(-\sqrt{-1}\ \frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\lambda F_{-k}}{\left[1+(\alpha+\beta+r+\lambda)\frac{\Delta\tau}{2}\right]-\frac{\Delta\tau}{2}\beta\exp\left(\sqrt{-1}\ \frac{2\pi}{N}k\right)-\frac{\Delta\tau}{2}\alpha\exp\left(-\sqrt{-1}\ \frac{2\pi}{N}k\right)-\frac{\Delta\tau}{2}\lambda F_{-k}}. \quad \text{(C.19)}$$

Recalling (C.7), it follows that

$$\begin{aligned}\alpha+\beta+r+\lambda &= \frac{\sigma^2}{2\Delta x^2}-\frac{\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{2\Delta x}+\frac{\sigma^2}{2\Delta x^2}+\frac{\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{2\Delta x}+r+\lambda\\ &= \frac{\sigma^2}{\Delta x^2}+r+\lambda,\end{aligned}$$

and

$$\begin{aligned}\Delta\tau\beta\exp\left(\sqrt{-1}\ \frac{2\pi}{N}k\right)+\Delta\tau\alpha\exp\left(-\sqrt{-1}\ \frac{2\pi}{N}k\right) = \\ \frac{\sigma^2\Delta\tau}{2\Delta x^2}\left[\exp\left(\sqrt{-1}\ \frac{2\pi}{N}k\right)+\exp\left(-\sqrt{-1}\ \frac{2\pi}{N}k\right)\right]+ \\ \frac{\Delta\tau\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{2\Delta x}\times\left[\exp\left(\sqrt{-1}\ \frac{2\pi}{N}k\right)+\exp\left(-\sqrt{-1}\ \frac{2\pi}{N}k\right)\right] \\ = \frac{\sigma^2\Delta\tau}{\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)+\frac{\sqrt{-1}\ \Delta\tau\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)}{\Delta x}\sin\left(\frac{2\pi}{N}k\right).\end{aligned}$$

Using the above results in (C.19), we find

$$C_k = \frac{\left[1-\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\frac{\Delta\tau}{2}\right]+\frac{1}{2}\left[\frac{\sigma^2\Delta\tau}{\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)+\sqrt{-1}\ \frac{\Delta\tau}{\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\Delta\tau\lambda F_{-k}\right]}{\left[1+\frac{1}{2}\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\Delta\tau\right]-\frac{1}{2}\left[\frac{\sigma^2\Delta\tau}{\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)+\sqrt{-1}\ \frac{\Delta\tau}{\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\Delta\tau\lambda F_{-k})\right]}. \tag{C.20}$$

Letting

$$\begin{aligned}F^R_{-k} &= \mathrm{Re}(F_{-k})\\ F^I_{-k} &= \mathrm{Im}(F_{-k}),\end{aligned}$$

equation (C.20) gives

$$|C_k|^2 = \frac{\left[1-\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\frac{\Delta\tau}{2}+\frac{\sigma^2\Delta\tau}{2\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)+\frac{\Delta\tau}{2}\lambda F^R_{-k}\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}{\left[1+\left(\frac{\sigma^2}{\Delta x^2}+r+\lambda\right)\frac{\Delta\tau}{2}-\frac{\sigma^2\Delta\tau}{2\Delta x^2}\cos\left(\frac{2\pi}{N}k\right)-\frac{\Delta\tau}{2}\lambda F^R_{-k}\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}, \tag{C.21}$$

or

$$|C_k|^2 = \frac{\left[1-\frac{r\Delta\tau}{2}-\frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1-\cos\left(\frac{2\pi}{N}k\right)\right)-\frac{\Delta\tau\lambda}{2}\left(1-F^R_{-k}\right)\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}{\left[1+\frac{r\Delta\tau}{2}+\frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1-\cos\left(\frac{2\pi}{N}k\right)\right)+\frac{\Delta\tau\lambda}{2}\left(1-F^R_{-k}\right)\right]^2+\left[\frac{\Delta\tau}{2\Delta x}\left(r-\lambda\kappa-\frac{1}{2}\sigma^2\right)\sin\left(\frac{2\pi}{N}k\right)+\lambda\frac{\Delta\tau}{2}F^I_{-k}\right]^2}. \tag{C.22}$$

Note that

$$F_{-k} = \frac{X_N}{N}\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}}\overline{f}_j\exp\left(\sqrt{-1}\,\frac{2\pi}{N}kj\right). \tag{C.23}$$

Then, from (3.22), we have

$$\frac{X_N}{N}\sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}}\overline{f}_j \le 1, \tag{C.24}$$

so that

$$|F_{-k}| \le 1, \tag{C.25}$$

and hence

$$-1 \le F^R_{-k} \le +1. \tag{C.26}$$

It then follows that $(\forall k \in -N/2+1, \ldots, +N/2)$

$$\left|1 + \frac{r\Delta\tau}{2} + \frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1 - \cos\left(\frac{2\pi}{N}k\right)\right) + \frac{\Delta\tau\lambda}{2}\left(1 - F^R_{-k}\right)\right| \geq \left|1 - \frac{r\Delta\tau}{2} - \frac{\sigma^2\Delta\tau}{2\Delta x^2}\left(1 - \cos\left(\frac{2\pi}{N}k\right)\right) - \frac{\Delta\tau\lambda}{2}\left(1 - F^R_{-k}\right)\right|, \tag{C.27}$$

and consequently $|C_k| < 1, \forall k$. As a result the scheme is unconditionally strictly stable.

Appendix D

American Options Under Jump Diffusion

In the following, we consider the problem of convergence of the iteration scheme (D.1). Convergence is proved by a number of properties of the intermediate quantities $\hat{V}^k$.

Algorithm D.1 Fixed point iteration scheme to solve the free boundary problem under the diffusion process.

Let $(V^{n+1})^0 = V^n$
Let $\hat{V}^k = (V^{n+1})^k$
Let $\hat{P}^k = P((V^{n+1})^k)$
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Solve

$$\left[I - (1-\theta)L + \hat{P}^k\right] \hat{V}^{k+1} = [I + \theta L]\, V^n + \hat{P}^k V^* + (1-\theta)\lambda\Delta\tau B\hat{V}^k + \theta\lambda\Delta\tau B V^n$$

  if $\max_i \frac{|\hat{V}_i^{k+1} - \hat{V}_i^k|}{\max(1, |\hat{V}_i^{k+1}|)} <$ tolerance **then**
    Quit
  end if
end for

Lemma D.1 (Bounded iterates). *Suppose that $\alpha_i, \beta_i \geq 0$ for all i in the discretization* (3.51) *and that B has the properties* (3.54)*. Then, for a given timestep, all iterates $\hat{V}^{k+1}$*

in scheme (D.1) *are bounded independent of* k.

Proof. Writing iteration (D.1) in component form gives

$$\begin{aligned}\left[1+(1-\theta)(\alpha_i+\beta_i+\lambda+r)\Delta\tau+\hat{P}_{ii}^k\right]\hat{V}_i^{k+1} &= \\ c_i+\hat{P}_{ii}^k V_i^*+(1-\theta)\lambda\Delta\tau\sum_j b_{ij}\hat{V}_j^k+(1-\theta)\Delta\tau\left[\alpha_i\hat{V}_{i-1}^{k+1}+\beta_i\hat{V}_{i+1}^{k+1}\right], &\qquad (D.1)\end{aligned}$$

where

$$c_i=\left(\left[I+\theta L\right]V^n+\theta\lambda\Delta\tau BV^n\right)_i. \qquad (D.2)$$

From the component form (D.1), it follows that $(i<imax)$

$$\begin{aligned}\left[1+(1-\theta)(\alpha_i+\beta_i+\lambda+r)\Delta\tau+\hat{P}_{ii}^k\right]|\hat{V}_i^{k+1}| &\le \|c\|_\infty+\hat{P}_{ii}^k\|V^*\|_\infty+(1-\theta)\lambda\Delta\tau\|\hat{V}^k\|_\infty \\ &\quad +(1-\theta)\Delta\tau\left[\alpha_i+\beta_i\right]\|\hat{V}^{k+1}\|_\infty. \qquad (D.3)\end{aligned}$$

Let m be an index such that

$$|\hat{V}_m^{k+1}| = \max_i|\hat{V}_i^{k+1}|=\|\hat{V}^{k+1}\|_\infty. \qquad (D.4)$$

Note that if $m=imax$ then we have

$$\|\hat{V}^{k+1}\|_\infty = |V_{imax}^n|\le\|V^n\|_\infty. \qquad (D.5)$$

Assume now that $m<imax$. Then from equations (D.3) and (D.4) we obtain

$$\left[1+(1-\theta)(\lambda+r)\Delta\tau+\hat{P}_{mm}^k\right]\|\hat{V}^{k+1}\|_\infty\le\|c\|_\infty+\hat{P}_{mm}^k\|V^*\|_\infty+(1-\theta)\lambda\Delta\tau\|\hat{V}^k\|_\infty. \qquad (D.6)$$

Equation (D.6) then gives

$$\|\hat{V}^{k+1}\|_\infty \leq \frac{\|c\|_\infty + P^k_{mm}\|V^*\|_\infty}{1+(1-\theta)(\lambda+r)\Delta\tau + P^k_{mm}} + \frac{(1-\theta)\lambda\Delta\tau\|\hat{V}^k\|_\infty}{1+(1-\theta)(\lambda+r)\Delta\tau + P^k_{mm}}. \tag{D.7}$$

Let

$$C_1 = \max(\|c\|_\infty, \|V^*\|_\infty) \text{ and } C_2 = \frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(\lambda+r)\Delta\tau} \tag{D.8}$$

so that

$$\|\hat{V}^{k+1}\|_\infty \leq C_1 + \frac{(1-\theta)\lambda\Delta\tau\|\hat{V}^k\|_\infty}{1+(1-\theta)(\lambda+r)\Delta\tau + P^k_{mm}} \leq C_1 + C_2\|\hat{V}^k\|_\infty. \tag{D.9}$$

Summing over the index k, equation (D.9) gives

$$\|\hat{V}^{k+1}\|_\infty \leq C_1\sum_{i=0}^{k} C_2^i + C_2^{k+1}\|\hat{V}^0\|_\infty. \tag{D.10}$$

Noting that $\hat{V}^0 = V^n$ and that $C_2 < 1$, equation (D.10) then gives

$$\|\hat{V}^{k+1}\|_\infty \leq \|V^n\|_\infty + \frac{C_1}{1-C_2}, \tag{D.11}$$

where C_1, C_2 are independent of k. From equation (D.5) we see that bound (D.11) is also valid for $m = imax$ and therefore for all m. □

After some manipulation, we can write iteration (D.1) as

$$\left[I - (1-\theta)L + \hat{P}^k\right](\hat{V}^{k+1} - \hat{V}^k) = (\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) + (1-\theta)\lambda\Delta\tau B(\hat{V}^k - \hat{V}^{k-1}). \tag{D.12}$$

In order to prove convergence of the scheme (D.1), it will be convenient to determine the sign of $(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k)$ in equation (D.12).

Lemma D.2 (Positive penalty term). *Given the definition of the penalty matrix $\hat{P}^k$ from equation (3.63), and the iteration scheme (D.1), we have that*

$$(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) \quad \geq \quad 0 \quad \textit{for all } k \geq 1 \ . \tag{D.13}$$

Proof. For each index i we have two possible cases. If $\hat{V}_i^k < V_i^*$ for component i then $\hat{P}_{ii}^k = \mathit{Large}$ so that

$$(\hat{P}_{ii}^k - \hat{P}_{ii}^{k-1})(V^* - \hat{V}^k)_i = (\mathit{Large} - \hat{P}_{ii}^{k-1})(V^* - \hat{V}^k)_i \geq 0.$$

On the other hand if $\hat{V}_i^k \geq V_i^*$ then $\hat{P}_{ii}^k = 0$ hence

$$(\hat{P}_{ii}^k - \hat{P}_{ii}^{k-1})(V^* - \hat{V}^k)_i = -\hat{P}_{ii}^{k-1}(V^* - \hat{V}^k)_i \geq 0.$$

Thus for all $k \geq 1$ we always have

$$(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) \geq 0. \tag{D.14}$$

□

Recall that an *M-matrix* has positive diagonals, non-positive off-diagonals, the row sums are non-negative with at least one such sum being positive. Such a matrix has the useful property that all the entries in its inverse are non-negative.

Lemma D.3 (M-matrices). *Let L, B and $\hat{P}^k$ be given by* (3.51), (3.52) *and* (3.63), *respectively. Assume that $\alpha_i \geq 0, \beta_i \geq 0$ in equation* (3.51), *that B has the properties* (3.54)

and that we use a Dirichlet boundary condition in (3.65)*. Then both*

$$[I-(1-\theta)L+P^k] \text{ and } [I-(1-\theta)L+P^k-(1-\theta)\lambda\Delta\tau B] \tag{D.15}$$

are M matrices.

Proof. It follows from equations (3.51), (3.52) and (3.52) that both of the above matrices have positive diagonals, non-positive off-diagonals and with row sum non-negative. Since a Dirichlet condition is imposed at $i = imax$ (3.65), for both matrices there is at least one row which has a strictly positive row sum. □

Recall that the discrete equations can be written as ($\theta = \theta_J$)

$$[I-(1-\theta)L+\hat{P}^{n+1}-(1-\theta)\lambda\Delta\tau B]V^{n+1} = [I+\theta L]V^n+\theta\lambda\Delta\tau BV^n+\hat{P}^{n+1}V^*. \tag{D.16}$$

We can now prove the following result:

Theorem D.1 (Uniqueness of solution). *Under the conditions required for Lemmas D.2 and D.3, any solution to equation* (D.16) *for a given timestep is unique.*

Proof. Suppose that we have two solutions V_1, V_2 to equation (D.16). Let $\hat{P}_1 \equiv P(V_1)$ and $\hat{P}_2 \equiv P(V_2)$ so that

$$[I-(1-\theta)L+\hat{P}_1-(1-\theta)\lambda\Delta\tau B]V_1 = [I+\theta L]V^n+\theta\lambda\Delta\tau BV^n+\hat{P}_1V^* \tag{D.17}$$

and

$$[I-(1-\theta)L+\hat{P}_2-(1-\theta)\lambda\Delta\tau B]V_2 = [I+\theta L]V^n+\theta\lambda\Delta\tau BV^n+\hat{P}_2V^*. \tag{D.18}$$

Equation (D.17) can be written as

$$[I-(1-\theta)L+\hat{P}_2-(1-\theta)\lambda\Delta\tau B]V_1+(\hat{P}_1-\hat{P}_2)V_1=[I+\theta L]V^n+\theta\lambda\Delta\tau BV^n+\hat{P}_1V^* \tag{D.19}$$

which after subtracting (D.18) from D.19 gives

$$[I-(1-\theta)L+\hat{P}_2-(1-\theta)\lambda\Delta\tau B](V_1-V_2) \quad = \quad (\hat{P}_1-\hat{P}_2)(V^*-V_1)\ . \tag{D.20}$$

Using the same arguments as in the proof of Lemma D.2 we have that $(\hat{P}_1-\hat{P}_2)(V^*-V_1)\geq 0$. From Lemma D.3 it follows that $I-(1-\theta)L+\hat{P}_2-(1-\theta)\lambda\Delta\tau B$ is an M-matrix and hence $(V_1-V_2)\geq 0$. Interchanging subscripts, we also have that $(V_2-V_1)\geq 0$ and hence $V_1=V_2$. □

Before we prove our main convergence result, we need the following Lemma.

Lemma D.4 (Norm of an iteration matrix). *Let A, B and $\hat{P}^k$ be given by* (3.51), (3.52) *and* (3.63), *respectively. Assume that $\alpha_i\geq 0, \beta_i\geq 0$ in equation* (3.51), *that B has the properties* (3.54) *and that we use a Dirichlet boundary condition in* (3.65). *Then for $\mathcal{O}^k=[I-(1-\theta)L+P^k]$ we have*

$$\|[\mathcal{O}^k]^{-1}B\|_\infty \quad \leq \quad \frac{1}{1+(1-\theta)(r+\lambda)\Delta\tau}\ . \tag{D.21}$$

Proof. Let y,z be vectors, z arbitrary, satisfying $\mathcal{O}^k y=Bz$. Then in component form we have that $y_{imax}=0$ and for $i<imax$:

$$[1+(1-\theta)(\alpha_i+\beta_i+r+\lambda)\Delta\tau+\hat{P}^k_{ii}]y_i=(1-\theta)\alpha_i\Delta\tau y_{i-1}+(1-\theta)\beta_i\Delta\tau y_{i+1}+\sum_j b_{ij}z_j. \tag{D.22}$$

From the properties of $\alpha_i, \beta_i, \hat{P}^k, B$, we then immediately have that

$$\|y\|_\infty \leq \frac{\|z\|_\infty}{1 + (1-\theta)(r+\lambda)\Delta\tau}, \tag{D.23}$$

giving (D.21). □

We are now in a position to prove our main convergence result

Theorem D.2 (Convergence of iteration (D.1)). *Let A, B and $\hat{P}^k$ be given by* (3.51), (3.52) *and* (3.63), *respectively. Assume that $\alpha_i \geq 0, \beta_i \geq 0$ in equation* (3.51), *that B has the properties* (3.54) *and that we use a Dirichlet boundary condition in* (3.65). *Then the iteration Algorithm D.1 is globally convergent to the unique solution of equation* (D.16) *for any initial iterate $\hat{V}^0$.*

Proof. Iteration (D.1) can be written as

$$\mathcal{O}^k(\hat{V}^{k+1} - \hat{V}^k) = (\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) + (1-\theta)\lambda\Delta\tau B(\hat{V}^k - \hat{V}^{k-1}), k \geq 1, \tag{D.24}$$

where $\mathcal{O}^k \equiv I - (1-\theta)A + \hat{P}^k$. For any $k \geq 1$ we can then write

$$(\hat{V}^{k+1} - \hat{V}^k) = U^k + W^k \cdot (\hat{V}^1 - \hat{V}^0), \tag{D.25}$$

with

$$\begin{aligned} U^k &= [\mathcal{O}^k]^{-1}(\hat{P}^k - \hat{P}^{k-1})[V^* - \hat{V}^k] \\ &\quad +(1-\theta)\lambda\Delta\tau[\mathcal{O}^k]^{-1}B[\mathcal{O}^{k-1}]^{-1}(\hat{P}^{k-1} - \hat{P}^{k-2})[V^* - \hat{V}^{k-1}] \\ &\quad + \ \ldots \\ &\quad +[(1-\theta)\lambda\Delta\tau]^{k-1}[\mathcal{O}^k]^{-1}B[\mathcal{O}^{k-1}]^{-1}B \ \ldots \ [\mathcal{O}^1]^{-1}B(\hat{P}^1 - \hat{P}^0)[V^* - \hat{V}^1], \\ W^k &= [(1-\theta)\lambda\Delta\tau]^k[\mathcal{O}^k]^{-1}B[\mathcal{O}^{k-1}]^{-1}B \ \ldots \ [\mathcal{O}^1]^{-1}B. \end{aligned} \tag{D.26}$$

We show that both U^k and W^k tend to zero as k gets large.

Note first that both $U^k \geq 0$ and $W^k \geq 0$. To show this we have that Lemma D.2 implies $(\hat{P}^k - \hat{P}^{k-1})(V^* - \hat{V}^k) \geq 0$ for all $k \geq 1$ while from Lemma D.3, we have that $[\mathcal{O}^k]^{-1} \geq 0$. Since $B \geq 0$ we have that all the components in U^k are non-negative. A similar statement is true for W^k since $\mathcal{O}^k$ is an M-matrix and since $B \geq 0$.

From Lemma D.4 and equation (D.26) we have that for each i

$$\|W^i\|_\infty \leq \left[\frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(r+\lambda)\Delta\tau}\right]^i, \tag{D.27}$$

and hence

$$\|\sum_{i=1}^{k} W^i\|_\infty \leq \sum_{i=1}^{k}\left[\frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)(r+\lambda)\Delta\tau}\right]^i \leq \left[\frac{(1-\theta)\lambda\Delta\tau}{1+(1-\theta)r\Delta\tau}\right]. \tag{D.28}$$

Thus $\{\sum_{i=1}^{k} W^i\}_{k=1,\ldots}$ is a sequence of non-decreasing terms which are bounded from above. As such, the sequence converges. In particular we have that W^k tends to zero as k tends to infinity.

Summing over the index k equation (D.25) gives

$$\hat{V}^{k+1} = \hat{V}^1 + \sum_{i=1}^{k} U^i + \sum_{i=1}^{k} W^i \cdot (\hat{V}^1 - \hat{V}^0). \tag{D.29}$$

From equation (D.27) we have that $(\sum_{i=1}^{k} W^i \cdot (\hat{V}^1 - \hat{V}^0)\ )$ converges to a finite value, furthermore from Lemma D.1 the left hand side of equation (D.29) is bounded from above. Thus the sequence $\{\sum_{i=1}^{k} U^i\}_{k=1,\ldots}$ is both non-decreasing and bounded from above. Hence this sequence also converges and so U^k approaches zero as k approaches infinity.

Thus a convergent limit exists, and from Theorem D.1, this is the unique solution to equation (D.16). □

Remark D.1 (Monotonicity of convergence). *Previous convergence results for penalty methods have typically required that the quantities V^k are monotonic (cf. [44]). From equation (D.25) we see that we do not necessarily have this property if $\hat{V}^1 < \hat{V}^0$. Of course we could ensure monotonicity by forcing $\hat{V}^1 \geq \hat{V}^0$. However the proof of Theorem D.2 shows that this is not really required. In addition, numerical experiments demonstrate that forcing monotonicity does not improve convergence.*

Remark D.2 (Speed of convergence). *Typically we have that $\lambda\Delta\tau \ll 1$. For example, for S&P 500 data, $\lambda \simeq .1$ [5], and a typical timestep is $\Delta\tau < .1$, giving $\lambda\Delta\tau \simeq .01$. For $\lambda\Delta\tau \ll 1$, equation (D.27) becomes*

$$\|W^i\|_\infty \quad \simeq \quad ((1-\theta)\lambda\Delta\tau)^i\,,$$

so that the term

$$\sum_{i=1}^{k} W^i \cdot (\hat{V}^1 - \hat{V}^0),$$

in equation (D.29) converges very rapidly. Experience with the penalty method for American

options with Brownian motion [44] (no jumps) indicates that the term

$$\sum_{i=1}^{k} U^i,$$

in equation (D.29) also converges rapidly. This rapid convergence will be confirmed with the numerical examples in Chapter 4.

Remark D.3 (Non M-matrices). *Our proof of convergence relies on the fact that the discretization of the PDE resulted in an M-matrix. However, we have observed (experimentally) that convergence is still rapid even if the coefficient matrix is not an M-matrix.*

An example where our discretization is not an M-matrix appears naturally as follows. It is often convenient to impose an asymptotic linearity boundary condition [89]

$$V_{SS} \quad = \quad 0 \quad ; \quad S \to \infty. \tag{D.30}$$

This boundary condition is particularly useful in complex path dependent cases where it is difficult to determine the asymptotic form of the solution [90].

Condition (D.30) is enforced by setting $(V_{SS})_i^n = 0$ at $i = imax$, and using a backward difference approximation for V_S. A little thought shows that for $r > 0$, this corresponds to using downwind weighting of the first order term at $i = imax$. This method can be shown to be stable [91], and the matrix solution can be obtained using Gaussian elimination without pivoting as long as the order of elimination is $i = 0, 1, \ldots$. However, in this case the coefficient matrix is no longer an M-matrix. In fact standard iterative methods for obtaining complementarity solutions may fail due to a small pivot since these methods [25] require repeated elimination steps in the forward ($i = 0, 1, \ldots$) and reverse ($i = imax, imax - 1, \ldots$) directions.

Appendix E

Fast Gauss Transform

The basics of the fast Gauss transform (FGT) algorithm (see [50]) are presented as an alternative to the fast Fourier transform (FFT) method to evaluate

$$I(S) = \int_0^\infty V(S\eta)g(\eta)d\eta. \tag{E.1}$$

This algorithm is of particular interest since it only requires $O(N)$ work, where N is the number of points used to approximate (E.1) by means of a quadrature formula and does not require a uniformly spaced grid. In comparison, the fast Fourier transform algorithm would require $O(N \log N)$ work as well as a uniform grid. While we limit our description to the one dimensional case, the FGT algorithm can be extended to higher dimensions (see [50]).

The basis of the fast Gauss transform algorithm is that the *Gauss transform* integral of a function f as given by

$$G(x) = \int f(y)e^{-\frac{(x_i-y_j)^2}{\omega}}dy, \tag{E.2}$$

can be approximated by

$$G_i \approx \sum_{j=0}^{N+1} q_j e^{-\frac{(x_i-y_j)^2}{\omega}}, \tag{E.3}$$

where q_j depends on both the weights of the chosen quadrature formula as well as $f(y_j)$, and $\omega > 0$. Equation (E.3) can then be transformed into an infinite series in terms of Hermite functions as follows

$$\begin{aligned} G_i &= \sum_{j=0}^{N+1} q_j \sum_{m\geq 0}\sum_{n\geq 0} \frac{1}{m!}\frac{1}{n!}\left(\frac{y_j-y_0}{\sqrt{\omega}}\right)^n h_{n+m}\left(\frac{x_0-y_0}{\sqrt{\omega}}\right)\left(\frac{x_i-x_0}{\sqrt{\omega}}\right)^m, \\ &= \sum_{m\geq 0}\frac{1}{m!}\left(\frac{x_i-x_0}{\sqrt{\omega}}\right)^m \sum_{n\geq 0} h_{n+m}\left(\frac{x_0-y_0}{\sqrt{\omega}}\right)\left[\frac{1}{n!}\sum_{j=0}^{N+1} q_j\left(\frac{y_j-y_0}{\sqrt{\omega}}\right)^n\right] \end{aligned} \tag{E.4}$$

where the Hermite function $h_n(x)$ is defined by

$$h_n(x) = (-1)^n \left(\frac{d}{dx}\right)^n e^{-x^2}. \tag{E.5}$$

Since the expansion series given by (E.4) converges very quickly [50], it is possible to determine the number of terms in the infinite series that are necessary to ensure that the truncation error meets an established tolerance depending on the quadrature weights (see [50] for details). Taking advantage of this fact and that $e^{-(x-y)^2}$ decreases very rapidly, L. Greengard and J. Strain [50] devised the fast Gauss transform algorithm to evaluate (E.2) in $O(N)$ time.

For our financial pricing problem, it is possible to write (E.1) as

$$I(x) = \int_{-\infty}^{\infty} \overline{V}(y) e^{\frac{-(x-y)^2}{\omega}} dy, \tag{E.6}$$

where $x = \log(S)$, $\omega > 0$ and $\overline{V}(y) = V(e^y)$. Note that this transformation is only obtainable for some probability density functions (e.g. [66] and [55]).

Table E.1 Input data used to value a European call option under the lognormal jump diffusion process. These parameters are approximately the same as those reported in [5] while fitting European call options on the S&P 500 stock index in April of 1999.

σ	0.15	r	0.05
γ	0.45	ν	0.0
λ	0.10	T	0.25
K	100.00		

In the following, we compare the fast Fourier transform approach with the fast Gauss transform approach. We use the implementation of the algorithm from [50]. We compare speed and accuracy of the FFT and FGT methods for a European put option while using the lognormal probability density function described in [66]. All of our tests were performed on a Sun ultra Sparc machine using the SUN C++ version 5.2 compiler and the F77 version 4.2 Fortran compiler. The input data is given by Table E.1.

Table E.2 Value of a European put option using Crank-Nicolson with constant timestepping with the FFT method to evaluate the integral. The input parameters are provided in Table E.1. The convergence ratio R is defined in equation (4.2). The exact solution is 2.781578 at $S = 100$. The number of points used for the FFT grid is 2^α, where α is the smallest integer such that the number of nodes in the non-uniform S grid $N \leq 2^\alpha$. Quadratic interpolation is used. The CPU time reported represents the average processing time of 10 consecutive runs when pricing a European option on a given grid size.

Size of S grid	No. of timesteps	CPU time (sec.)	$S = 100$	
			Value	R
128	25	0.036	2.778662	n.a.
255	50	0.095	2.780837	n.a.
509	100	0.292	2.781387	3.953
1017	200	1.138	2.781530	3.857
2033	400	4.496	2.781566	3.917
4065	800	18.172	2.781575	3.972

We begin by comparing the accuracy of the solution between the FFT method (see

Table E.2) and the FGT method (see Table E.3). Three different quadrature approaches are used for the FGT method, that is trapezoidal, Simpson and Simpson 3/8 rules. We notice that the accuracy of the solution is relatively unaffected by the choice of the quadrature rule independently of the tolerance used. For sufficiently small tolerance, the exact value of the European put option defined by the input parameters in Table E.1 is recovered. With 255 points and a tolerance equal to 10^{-4}, penny accuracy is obtained.

Next, we compare the speed and accuracy of the FGT and FFT methods. The CPU time reported in tables E.2 and E.4 represents the average processing time of 10 consecutive runs when pricing a European option on a given grid size. For the FGT method we use the trapezoidal quadrature rule (see Table E.3). In table E.2, we observe quadratic convergence to the exact solution when the FFT algorithm is used. Similarly for the FGT algorithm (see Table E.4), quadratic convergence is also recovered if a sufficiently small tolerance (i.e. $tol = 10^{-6}$) is chosen. Any tolerance larger than $tol > 10^{-6}$ did not result in quadratic convergence.

In Table E.5, we report the CPU time for different tolerances for the FGT algorithms. Before going any further, we must emphasize that the FFT algorithm requires a uniform $x = \log(S)$ grid, while the FGT algorithm does not. Thus, for each fixed point iteration two interpolation steps are required for the FFT while none are necessary for the FGT. As such, we could expect the FGT to be a better method to approximate equation (E.6), assuming that the constant of the running time complexity is not too large.

However, we find that the FFT method is much faster than the FGT approach (see Table E.5). From an option pricing perspective, it takes one tenth of a second to obtain penny accuracy when using the FFT algorithm, while it takes approximately half a second for the FGT algorithm with a tolerance of 10^{-4}.

Based on these observations, we conclude that the fast Fourier transform is a more appropriate choice for our particular problems. Further experiments lead us to determine

Table E.3 Value of a European put option using Crank-Nicolson with constant timestepping with the FGT algorithm to evaluate the integral (E.6). Convergence tests are presented for different tolerance values and integration schemes. The input parameters are provided in Table E.1. The exact solution is 2.781578 at $S = 100$.

Size of S grid	Trapezoidal rule	Simpson's rule	Simpson's 3/8 rule
128	2.778658	2.777844	2.779892
255	2.794691	2.795355	2.794930
509	2.795703	2.795660	2.795670
1017	2.795905	2.795931	2.795903
2033	2.796220	2.796150	2.796242
4065	2.796121	2.796126	2.796113

(A) $tol = 10^{-2}$.

Size of S grid	Trapezoidal rule	Simpson's rule	Simpson's 3/8 rule
128	2.778662	2.777849	2.779894
255	2.780078	2.780103	2.780171
509	2.780611	2.780610	2.780613
1017	2.780749	2.780749	2.780749
2033	2.780779	2.780780	2.780779
4065	2.780790	2.780789	2.780790

(B) $tol = 10^{-4}$.

Size of S grid	Trapezoidal rule	Simpson's rule	Simpson's 3/8 rule
128	2.778662	2.777849	2.779894
255	2.780837	2.780875	2.780937
509	2.781418	2.781417	2.781420
1017	2.781561	2.781561	2.781561
2033	2.781598	2.781598	2.781598
4065	2.781607	2.781607	2.781607

(C) $tol = 10^{-6}$.

Size of S grid	Trapezoidal rule	Simpson's rule	Simpson's 3/8 rule
128	2.778662	2.777849	2.779894
255	2.780837	2.780875	2.780937
509	2.781385	2.781385	2.781388
1017	2.781528	2.781528	2.781528
2033	2.781565	2.781565	2.781565
4065	2.781574	2.781574	2.781574

(D) $tol = 10^{-8}$.

Size of S grid	Trapezoidal rule	Simpson's rule	Simpson's 3/8 rule
128	2.778662	2.777849	2.779894
255	2.780837	2.780875	2.780937
509	2.781386	2.781385	2.781388
1017	2.781529	2.781529	2.781529
2033	2.781565	2.781565	2.781565
4065	2.781575	2.781575	2.781575

(E) $tol = 10^{-10}$.

Size of S grid	Trapezoidal rule	Simpson's rule	Simpson's 3/8 rule
128	2.778662	2.777849	2.779894
256	2.780837	2.780875	2.780937
510	2.781387	2.781386	2.781389
1018	2.781530	2.781530	2.781529
2034	2.781566	2.781566	2.781566
4066	2.781575	2.781575	2.781575

(F) $tol = 10^{-12}$.

that for very large problems the fast Gauss transform algorithm was actually faster than the fast Fourier transform. However, the size at which $O(N)$ complexity is superior of $0(N \log N)$ is large. Nonetheless, we must emphasize that our FFT framework can be applied to arbitrary jump probability densities, while the FGT algorithm can only be applied to specific probability density functions.

Table E.4 Value of a European put option using Crank-Nicolson with constant timestepping with the FGT algorithm to evaluate the integral (E.6). Convergence tests are presented for different tolerance values. The tolerance value indicates at which point the infinite sums (E.4) must be truncated. The input parameters are provided in Table E.1. The convergence ratio R is defined in equation (4.2). The exact solution is 2.781578 at $S = 100$. The CPU time reported represents the average processing time of 10 consecutive runs when pricing a European option on a given grid size.

Size of S grid	CPU time (sec.)	$S = 100$	
		Value	R
128	0.188	2.778658	n.a.
255	0.36	2.794691	n.a
509	0.95	2.795703	5.006
1017	2.945	2.795905	0.640
2033	9.963	2.796220	0.640
4065	39.811	2.796121	-3.167

(A) $tol = 10^{-2}$.

Size of S grid	CPU time (sec.)	$S = 100$	
		Value	R
128	0.241	2.778662	n.a.
255	0.487	2.780078	n.a.
509	1.37	2.780611	2.660
1017	4.439	2.780749	3.841
2033	16.318	2.780779	4.676
4065	85.966	2.780790	2.793

(B) $tol = 10^{-4}$.

Size of S grid	CPU time (sec.)	$S = 100$	
		Value	R
128	0.317	2.778662	n.a.
255	0.656	2.780837	n.a.
509	1.936	2.781418	3.742
1017	6.491	2.781561	4.068
2033	33.186	2.781598	3.908
4065	155.798	2.781607	3.955

(C) $tol = 10^{-6}$.

Size of S grid	CPU time (sec.)	$S = 100$	
		Value	R
128	0.359	2.778662	n.a.
255	0.772	2.780837	n.a.
509	2.364	2.781385	3.965
1017	8.639	2.781528	3.845
2033	44.929	2.781565	3.919
4065	195.953	2.781574	3.948

(D) $tol = 10^{-8}$.

Size of S grid	CPU time (sec.)	$S = 100$	
		Value	R
128	0.367	2.778662	n.a.
255	0.727	2.780837	n.a.
509	2.463	2.781386	3.960
1017	11.065	2.781529	3.851
2033	55.282	2.781565	3.917
4065	239.711	2.781575	3.973

(E) $tol = 10^{-10}$.

Size of S grid	CPU time (sec.)	$S = 100$	
		Value	R
128	0.45	2.778662	n.a.
256	1.004	2.780837	n.a.
510	3.414	2.781387	3.953
1018	18.729	2.781530	3.857
2034	80.909	2.781566	3.917
4066	341.184	2.781575	3.971

(F) $tol = 10^{-12}$.

However, it should be pointed out that it may be possible to improve the implementation of [50], so that the FGT method could be more competitive with the FFT method. It is unlikely that the implementation of the FFT algorithm can be improved significantly.

Table E.5 CPU time comparison between the FFT and the FGT algorithms when pricing a European put option using Crank-Nicolson with constant timestepping for different tolerance values. The input parameters are provided in Table E.1. The CPU time reported represents the average processing time of 10 consecutive runs when pricing a European option on a given grid size.

Size of S grid	CPU time (sec.)						
	FFT	FGT					
		$tol = 10^{-2}$	$tol = 10^{-4}$	$tol = 10^{-6}$	$tol = 10^{-8}$	$tol = 10^{-10}$	$tol = 10^{-12}$
128	0.036	0.188	0.241	0.317	0.359	0.367	0.45
255	0.095	0.36	0.487	0.656	0.778	0.727	1.004
509	0.292	0.95	1.37	1.936	2.364	2.463	3.414
1017	1.138	2.945	4.439	6.491	8.639	11.065	18.729
2033	4.496	9.963	16.318	33.186	44.929	55.282	80.909
4065	18.172	39.811	85.966	155.798	195.953	239.711	341.184

Furthermore, it may be possible to devise a high accuracy quadrature rule with very few points such that penny accuracy can be obtained for grids with few points [4]. However, in this case it may be necessary to interpolate the solution $V(S, \tau)$ back and forth between two different grids, which may in turn affect the speed of the algorithm.

Appendix F

Non-uniform FFT

Recall that in order to evaluate the correlation integral (3.10), we used an FFT method. This requires that the nodal values $\overline{V}_i$ be equally spaced on a $\log S$ grid. However, since this is not efficient for solving the PDE, we interpolate from the PDE grid to an equally spaced $\log S$ grid. After evaluating the correlation integral, we interpolate from the equally spaced $\log S$ grid back to the original PDE grid.

However, there have been several recent papers concerning the problem of using Fourier methods on unequally spaced grids. There are two basic problems

Forward Given discrete values f_i, $i = -N/2+1, ..., +N/2$, then determine F_k, $k = -N/2+1, ..., +N/2$, such that

$$f_i = \frac{1}{X_N} \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}} F_l \exp(\sqrt{-1}\frac{2\pi}{X_N} x_i l), \tag{F.1}$$

where $X_N = x_N/2 - x_{-N/2+1}$.

Reverse Given discrete values F_k, $k = -N/2+1, ..., +N/2$, determine f_i, $i = -N/2+1, ..., +N/2$, such that

$$f_i = \frac{1}{X_N} \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}} F_l \exp(\sqrt{-1}\frac{2\pi}{X_N} x_i l), \tag{F.2}$$

Note that if the nodes x_i are at locations $x_i = iX_N/N$, then both the forward and reverse problems are easily computed using the discrete orthogonality conditions (C.14). However, if the x_i are arbitrary nodes, then a direct evaluation of the forward algorithm requires $O(N^3)$ operations, and the reverse algorithm requires $O(N^2)$ operations.

There are several algorithms for carrying out the Reverse problem. In [88], several methods are tested for computing an approximation to the Reverse problem (F.2). In fact, the straightforward method of simply computing

$$\hat{f}_j = \frac{1}{X_N} \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}} F_l \exp(\sqrt{-1}\frac{2\pi}{N} j l), \tag{F.3}$$

at $\hat{f}_j = f(jX_N/N)$ using an FFT, and then using Lagrange interpolation to determine the required values at $f(x_i)$, is quite competitive with other methods.

The Forward problem (F.1) can be posed as a linear algebra problem, i.e. if $f = [f_{-N/2+1}, ..., f_{N/2}]'$, and $F = [F_{-N/2+1}, ..., F_{N/2}]'$, then the Forward problem can be stated

as given the right hand side vector f, solve the system

$$AF = f \tag{F.4}$$

for F, where

$$[A]_{ij} = \frac{1}{X_N} \exp(\sqrt{-1}\frac{2\pi}{X_N} x_i j) \tag{F.5}$$

Several methods have been proposed for solving (F.4). The most efficient method appears to the technique suggested in [7] , which uses a preconditioned GMRES approach. However, as pointed out in [7], the pre-conditioner suggested results in rapid convergence only in cases where the x_i are small perturbations from a grid with $x_i = iX_N/N$. We have carried out numerical experiments, using the method in [7], and for the clustered grids which are typically used in option pricing, the number of GMRES iterations required for convergence is unacceptably large.

An alternative method for the Forward problem is suggested in [39]. This method uses a Fast Multipole method. However, in [76], the authors indicate that this method has problems unless the nodes are also almost equally spaced.

In any case, it should be recalled that the discrete approximation for the integral (3.10) is only second order correct. Consequently, we do not need to evaluate the Fourier coefficients to high accuracy.

Appendix G

Estimation of the Market Price of Risk

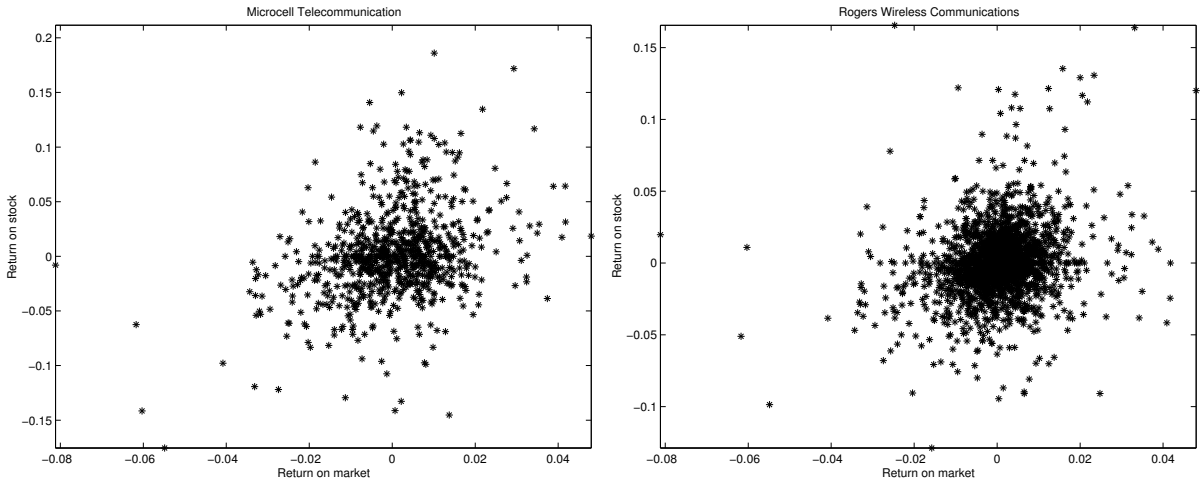
G.1 The Schwartz and Moon Approach

We follow the approach described in [82], which uses stock market data. However, Bell Mobility is not a publicly traded company, so we need to find public Canadian companies which are in the wireless communications business. Two such firms are Rogers Wireless Communications Inc. and Microcell Telecommunications Inc.

For each company, we must first compute the systematic risk exposure using the standard capital asset pricing model.[1] Because each firm has significant amounts of debt outstanding, we will initially use the "levered" equity beta β as the measure of the systematic risk. To estimate β, we run linear regressions of returns for each stock versus the return on the market. We use the Toronto Stock Exchange (TSE) 300 index as a proxy for the return on the market. We obtain daily total return data for the TSE 300 index and each firm from the Canadian Financial Markets Research Centre database. The sample for Microcell runs

[1] Readers unfamiliar with these concepts should consult a corporate finance text such as [81].

from October 9, 1997 to December 29, 2000. The sample for Rogers runs from August 9, 1991 through December 29, 2000. Figure G.1 provides plots of the return for the market versus the return for Microcell and Rogers.



(A) Microcell. (B) Rogers.

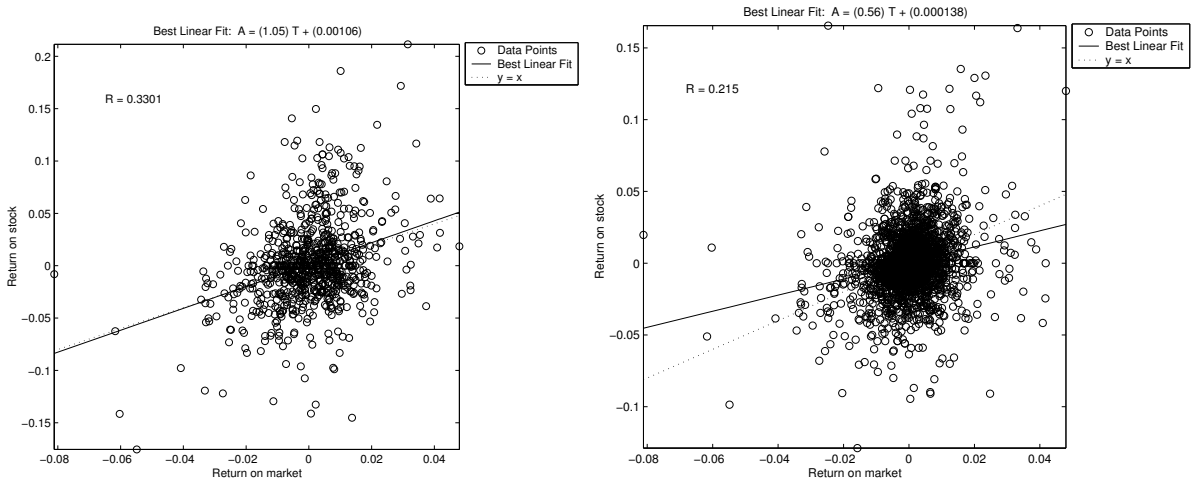
FIGURE G.1: *Daily return on TSE 300 index (horizontal axis) versus daily return on wireless telecommunications firm (vertical axis).*

Figure G.2 shows the line of best fit superimposed on each point representing pairs of daily return data. For Microcell, we find $\beta = 1.0455$, while $\beta = .5603$ for Rogers.

Having estimated the levered betas for both firms, we next undo the effects of leverage by calculating the beta for a hypothetical unlevered (i.e. no debt) version of each firm.[2] To compute the unlevered firm's beta from the levered equity beta, the market value of the firm's debt and equity must be estimated, along with its corporate tax rate. The unlevered firm's beta is then given by

$$\beta^{\text{unlevered}} = \frac{E}{E + (1 - T_c)D}\beta, \tag{G.1}$$

[2]Note that the calculations to follow involve some simplifying approximations commonly used in corporate finance (such as the assumption that each firm's debt is perpetual).



(A) Microcell. The slope of the regression line $\beta = 1.0455$. The regression $R^2 = .3301$.

(B) Rogers. The slope of the regression line $\beta = 0.5603$. The regression $R^2 = .2150$.

FIGURE G.2: *Regressions of daily returns for wireless communications firms on TSE 300 index.*

where

$$E = \text{market value of equity (total number of shares times share price)},$$

$$D = \text{market value of long term debt},$$

$$T_c = \text{corporate tax rate}.$$

As an initial approximation, for each company we will assume a tax rate of 40%. Tables G.1 and G.2 contain the information we use to calculate the unlevered beta for each company.

Using equation (G.1), we find that

$$\begin{aligned}\beta_{\text{Microcell}}^{\text{unlevered}} &= \frac{2.68 \times 240}{2.68 \times 240 + (1. - .4) \times 1887.048} \times 1.0455 \\ &= .37876,\end{aligned}$$

Table G.1 Microcell Telecommunications Inc. corporate data. Dollar figures are in Canadian funds. The data was obtained from `http://www.globeinvestor.com/` on February 13, 2002 and Microcell's quarterly financial reports.

Long term debt	\$1,887,048,000
Corporate tax rate	40%
Stock price (on February 13, 2002)	\$2.68
Number of shares	240,000,000

Table G.2 Rogers Wireless Telecommunications Inc. corporate data. Dollar figures are in Canadian funds. The data was obtained from `http://www.globeinvestor.com/` on April 18, 2002 and Rogers' quarterly financial reports.

Long term debt	\$2,305,638,000
Corporate tax rate	40%
Stock price (on April 18, 2002)	\$17.78
Number of shares	144,400,000

and

$$\begin{aligned}\beta_{\text{Rogers}}^{\text{unlevered}} &= \frac{17.78 \times 144.4}{17.78 \times 144.4 + (1. - .4) \times 2305.638} \times .5603 \\ &= .36411.\end{aligned}$$

These two values are remarkably close. Averaging them, we estimate that the unlevered Bell Mobility β is given by $\beta_{\text{Bell Mobility}}^{\text{unlevered}} = .3714$. Note that our use of unlevered betas means that our real option valuation is biased low for an investment project financed with debt, as interest tax shields have not been accounted for. The value of these tax shields could be added later, if desired. However, it might be argued that, given the current financial situation in the telecommunications sector, new debt financing is unlikely to be available at present.

Now that we have estimated the unlevered beta for Bell Mobility, we can compute the market price of risk ζ. As mentioned, we follow the methodology described in [82]. For readers unfamiliar with this approach, we provide a short and somewhat simplified description here.

Suppose there is a single stochastic factor X (in our context this is Q), which follows the risk-adjusted process:

$$dX = (\mu - \psi)Xdt + \sigma X dz, \tag{G.2}$$

where μ is the real world drift, σ is the volatility, ψ is the risk premium, and dz is the increment of a Wiener process. Note that the risk premium $\psi = \zeta\sigma$ (i.e. the market price of risk multiplied by the volatility). Let the firm's stock price be S. From Itô's lemma, we have:

$$\frac{dS}{S} = \frac{\left[\frac{1}{2}\sigma^2 X^2 S_{XX} + (\mu - \psi)XS_X + S_t\right]}{S}dt + \frac{\sigma X S_X}{S}dz, \tag{G.3}$$

where the risk premium is:

$$\frac{\psi X S_X}{S}. \tag{G.4}$$

The intertemporal capital asset pricing model (ICAPM) is then applied in the following way. The firm's stock β, denoted by β_S, depends on the covariance between returns on the market portfolio M and returns on the stock. This can be written as a function of the "β" of the stochastic factor X:

$$\beta_S = \frac{\sigma_{SM}}{\sigma_M^2} = \frac{XS_X}{S}\frac{\sigma_{XM}}{\sigma_M^2} = \frac{XS_X}{S}\beta_X, \tag{G.5}$$

where σ_{SM} is the covariance between changes in S and M and similarly for σ_{XM}. In the ICAPM, the expected return on the stock is:

$$r_S = r_f + \beta_S(r_M - r_f) = r_f + \frac{XS_X}{S}\beta_X(r_M - r_f) \tag{G.6}$$

where r_f denotes the risk free rate of interest and r_M is the expected return on the market portfolio. Equating the risk premium from (G.4) with that implied in (G.6) gives:

$$\begin{aligned} \frac{\psi X S_X}{S} &= \frac{X S_X}{S} \beta_X (r_M - r_f) \\ \Rightarrow \psi &= \beta_X (r_M - r_f). \end{aligned} \tag{G.7}$$

Using (G.5), we have:

$$\psi = \frac{S \beta_S}{X S_X} (r_M - r_f), \tag{G.8}$$

i.e. the risk premium is a function of the expected excess market return, the firm's current stock price, the β of the firm's stock price, the current level of the stochastic factor X, and S_X.

Returning to our context, we then have:

$$\psi = \zeta \sigma = \frac{S \beta^{unlevered}_{Bell\ Mobility}}{Q S_Q} (r_M - r_f), \tag{G.9}$$

where S is Bell Canada Enterprises (BCE)'s current stock price, Q is the current level of traffic, and S_Q is the first derivative of the stock price with respect to the level of traffic.

All the parameters from equation (G.9) are known (or have been estimated previously) except S_Q, r_M and r_f. For the risk free rate r_f, we assume a value of $r_f = .04$. We assume that the expected market return r_M is 6% higher than the risk free rate (consistent with the average level for the past 50 years of Canadian data). Thus we have $r_M = .10$. S_Q is a more challenging parameter since there is no direct data from which we can determine it. Consequently, to estimate S_Q, we run linear regressions of BCE's stock price on the traffic data for the various time series. Our estimates of S_Q are the slope coefficients of these regressions. We obtain $S_Q = 1.8656 \times 10^{-4}$ for time series A, 9.0057×10^{-5} for time series B, and 8.4996×10^{-5} for time series C.

On August 2, 2002, BCE's stock price was \$39.08. Combining this information with our estimated values of S_Q and using $Q = 1.9197 \times 10^5$ (the level of bouncing busy hour traffic on August 2, 2002) and equation (G.9) gives

$$\psi = \begin{cases} \frac{39.08 \times .3714}{1.9197\,10^5 \times 1.8656\,10^{-4}} \times (.1 - .04) = .0243 & \text{(Time series A)} \\ \frac{39.08 \times .3714}{1.9197\,10^5 \times 9.0057\,10^{-5}} \times (.1 - .04) = .0504 & \text{(Time series B)} \\ \frac{39.08 \times .3714}{1.9197\,10^5 \times 8.4996\,10^{-5}} \times (.1 - .04) = .0534 & \text{(Time series C)} \end{cases}$$

Hence, using the estimates of σ for each time series from Table 2.3 and the values of ψ calculated above, equation (G.9) gives estimates of $\zeta \approx .05$ (based on time series A), $\approx .07$ (based on time series B), and $\approx .17$ (based on time series C). Averaging these gives a value of around .10.

One of the key assumptions above was that both Microcell Telecommunications and Rogers Wireless Communications face corporate tax rates of 40%. This is potentially problematic, especially for Microcell, which has been in financial difficulty and may not be likely to be in a tax paying position. Repeating our calculations, but assuming that each firm has a tax rate of 0%, we obtain $\zeta \approx .04$ (based on time series A), $\approx .05$ (based on time series B), and $\approx .13$ (based on time series C), with an average of about .075. Any other combination of tax rates for the two firms lying between 0% and the statutory rate of 40% gives rise to estimates of ζ lying between the values calculated when both firms have rates of 0% and 40%.

G.2 An Alternative Approach to Estimate the Market Price of Risk

In this appendix we present a more traditional approach to estimate the market price of risk. This is described in standard texts such as [52]. It is based on the correlation between market returns (i.e. TSE 300 index) and changes in the bouncing busy hour traffic levels. Using this approach, it is possible to estimate the market price of risk as

$$\zeta = \rho\zeta_M, \tag{G.10}$$

where ρ is the correlation between the bouncing busy hour changes and market returns and ζ_M is the market price of risk for the stock market. Using the capital asset pricing model, the market price of risk is $\zeta_M = (r_M - r_f)/\sigma_M$, where r_M is the market return, r_f is the risk free rate and σ_M is the market volatility. We assume (as in Appendix G that $r_M - r_f = .06$. We calculate $\sigma_M = .2383$, using historical market data over the same time period as our network traffic data series. Over this period, we find estimates of ρ of .1367 for time series A, .1171 for time series B, and .1188 for time series C. Using equation (G.10) we obtain estimates for ζ of .0344 for time series A, .0295 for time series B, and .0299 for time series C. Consequently, our estimates of ζ range from a high value of $\approx .17$ (based on one particular time series in Appendix G) to a low of around .03 (based on all time series in Appendix G.1).

Appendix H

Estimation of the Jump Parameters for the Real Options Investment Management Problem

In this chapter, given wireless traffic data time series, an empirical method is developed to estimate the first two moments of the wireless traffic jump distribution. It is recognized that the approach presented here may not be complete, other approaches can be found in [1, 6, 38, 40, 49, 51] and references therein. However given the number of data points available, we believe that our procedure is sufficient to assess the effects of jumps in traffic on the optimal wireless investment decision problem (see Chapter 6).

In Chapter 2, we showed that a geometric Brownian motion (GBM) stochastic process, given by

$$dQ = \mu Q dt + \sigma Q dZ, \tag{H.1}$$

was a good fit to model traffic demand Q (μ represents the drift rate or growth rate, σ represents the volatility and dZ is the increment of a Gauss-Wiener process [52, 89]). When

estimating both the drift rate μ and the volatility σ, vacation periods and the qualitatively observed sudden jumps in traffic were smoothed out from the time series. Table H.1 presents a summary of estimates for the drift rate μ and the volatility σ.

Table H.1 Result summary table for different time series (different switches in Toronto). The Ljung-Box test is realized at the 5% significance level. The Ljung-Box test $H = 0$ indicates random data, i.e. equation (H.1) is a good fit to the data.

Time series	Drift rate μ %/$(year)$	Volatility (filtered data) σ %/$(year)^{\frac{1}{2}}$	Ljung-Box test
A	-24.75	90.69	0
B	40.84	74.7	0
C	72.46	32.74	0

We assume that the jumps J in wireless traffic are lognormally distributed [66] and are given by

$$g(J) = \frac{\exp\left(-\frac{(\log(J)-\nu)^2}{2\gamma^2}\right)}{\sqrt{2\pi}\gamma J}, \tag{H.2}$$

where ν corresponds to the mean and γ to the standard deviation. To estimate ν and γ we proceed as follows for each time series:

- **Step 1** Estimate the drift rate μ and volatility σ ignoring vacation periods and sudden jumps in traffic.
- **Step 2** Compute the logarithm of the traffic time series $y = \log(Q)$ ignoring vacation periods but including the sudden large jumps in traffic (i.e. the jumps that were ignored in Step 1). Using Ito's lemma [89], equation (H.1) becomes

$$dy = (\mu - \frac{\sigma^2}{2})dt + \sigma dZ. \tag{H.3}$$

- **Step 3** Let the spacing between sampling dates be Δt. Then the conditional distribution of y_t, given y_{t-1} is normal with mean $y_{t-1} + (\mu - 0.5\sigma^2)\Delta t$ and standard

deviation $\sigma\sqrt{\Delta t}$. Consequently,

$$z = \frac{y_t - [y_{t-1} + (\mu - 0.5\sigma^2)\Delta t]}{\sigma\sqrt{\Delta t}}, \tag{H.4}$$

z is normally distributed with mean zero and unit variance. Values of $|z|$ which deviates more than 1.94 standard deviations of the mean are assumed to be jumps in traffic. Basically, we are assuming that capacity usage follows a geometric Brownian motion process (H.1) (i.e. $P(-1.94 \leq z \leq 1.94) = .95$), and the jumps corresponds to extreme events $1 - P(-1.94 \leq z \leq 1.94)$. However, since only large deviations from the mean are considered, the standard deviation for the traffic jumps is certainly biased upward. The jumps frequency λ, the mean ν and the standard deviation γ are then estimated.

- **Step 4** Iterate back to **Step 2** taking into account the jumps in traffic that have just been detected. If no new jumps are detected terminate.

Table H.2 Jump parameters summary table for different time series (different switches in Toronto).

Time series	# iterations	Jump frequency per year	Jump mean ν	Jump standard deviation γ
A	1	5	-0.1284	0.3961
B	3	6	-0.0121	0.2304
C	4	6	-0.0119	0.0850
Average	n.a.	$\approx$ 5.6	$\approx$ -.0508	$\approx$ 0.2372

Table H.2 presents our jump parameter estimates. Note that for the wireless time series B and C, a larger number of jumps in traffic were discovered resulting in new values for the drift rate and lower volatility, see Tables H.1 and H.3. Since we are only examining jumps on a single day of each week, we estimate the number of jumps in traffic per year (occurring on business day) by multiplying the estimates in Table H.2 by five. Consequently, we have

Table H.3 Drift rate and volatility summary table for different time series (different switches in Toronto).

Time series	Drift rate μ %/(year)	Volatility σ per %/(year)$^{\frac{1}{2}}$
A	-24.75	90.69
B	21.72	41.91
C	69.35	21.22
Average	≈ 22.11	≈ 51.27

that $\lambda \approx 28$/year, for jumps which can occur on any business day.

Appendix I

Semi-Lagrangian Approach for Path Dependent Options

When pricing continuously observed Asian options, we use the same unequally spaced grid in the A and S directions, on the domain $[0, S_{imax}] \times [0, A_{jmax}]$, with $A_{jmax} = S_{imax}$. For completeness, the discretization grid that was used in Section 5.6 is presented below. Note that this grid needs to be scaled by the value of the strike of the option considered.

```
const int n = 53;
double Xarray[n] = {
  .0,   .10,  .20, .30, .40,.50,  .60,  .70, .75,    .80, .84,
  .86,  .88,  .90, .91, .92,  .93, .94, .95,.96,  .97,  .98, .99,
 1.00, 1.01, 1.02, 1.03, 1.04,  1.05, 1.06, 1.07, 1.08, 1.09,
 1.10,  1.12, 1.14, 1.16, 1.18, 1.20, 1.23,  1.26, 1.30, 1.40,
 1.50, 1.75,  2.00, 2.50, 3.00, 5.00, 10.00, 50.00, 250.00, 2.e+5};
```

When testing convergence, the number of points of the above grid is doubled (i.e. $2N-1$) at each iteration by inserting new nodes at grid midpoints.

Bibliography

[1] J. Ait-Sahalia and R. Kimmel. Estimating affine multifactor term structure models using closed-form likelihood expansions. Working Paper, Princeton University, 2002.

[2] A. Allievi and R. Bermejo. Finite element modified method of characteristics for the Navier-Stokes equation. *International Journal for Numerical Methods in Fluids*, 32:439–464, 2000.

[3] A. L. Amadori. The obstacle problem for nonlinear integro-differential equations arising in option pricing. Working paper, Istituto pre le Applicazioni del Calcolo "M. Picone", Rome, http://www.iac.rm.cnr.it/~amadori/papers/Ajdop00.ps.gz, 2000.

[4] K.I. Amin. Jump diffusion option valuation in discrete time. *Journal of Finance*, 48:1833–1863, 1993.

[5] L. Andersen and J. Andreasen. Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing. *Review of Derivatives Research*, 4:231–262, 2000.

[6] T.G. Andersen and J. Lund. Estimating continuous-time stochastic volatility models of the short-term interest rate. *Journal of Econometrics*, (77):343–377, 1997.

[7] C. Anderson and M.D. Dahleh. Rapid computation of the discrete Fourier transform. *SIAM Journal on Scientific Computing*, 17:913–919, 1996.

[8] J. Andreasen. The pricing of discretely sampled Asian and lookback options. *Journal of Computational Finance*, 2:5–30, Fall 1998.

[9] E. Ayache, P. Henrotte, and X. Wang. Can anyone solve the smile problem? Wilmott magazine, January 2004.

[10] G. Bakshi, C. Cao, and Z. Chen. Empirical performance of alternative option pricing models. *Journal of Finance*, 52:2003–2049, 1997.

[11] G. Barles. Convergence of numerical schemes for degenerate parabolic equations arising in finance. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 1–21. Cambridge University Press, Cambridge, 1997.

[12] J. Barraquand and T. Pudet. Pricing of American path-dependent contingent claims. *Mathematical Finance*, 6:17–51, 1996.

[13] D. S. Bates. Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche mark options. *Review of Financial Studies*, 9:69–107, 1996.

[14] D.S. Bates. The crash of '87: Was it expected? The evidence from options markets. *Journal of Finance*, 46:1009–1044, 1991.

[15] J. Becker. A second order backward difference method with variable timesteps for a parabolic problem. *BIT - Numerical Mathematics*, 38(4):644–662, 1998.

[16] R. Bermejo. Analysis of a class of quasi-monotone and conservative semi-Lagrangian advection schemes. *Numerische Mathematik*, 87:597–623, 2001.

[17] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–659, 1973.

[18] N. Borovykh and M.N. Spijker. Resolvent conditions and bounds on the powers of matrices, with relevance to the numerical stability of initial value problems. *Journal of Computational and Applied Mathematics*, 125:41–56, 2000.

[19] P.P. Boyle. New life forms on the option landscape. *Journal of Financial Engineering*, 2(3):217–252, 1996.

[20] P.P. Boyle and D. Emanuel. Options on the general mean. Working paper, University of British Columbia, 1980.

[21] M. Briani, C. La Chioma, and R. Natalini. Convergence of numerical schemes for viscosity solutions to integro-differential degenerate parabolic problems arising in financial theory. *Numerishe Mathematik*, forthcoming, 2003.

[22] M. Broadie and Y. Yamamoto. Application of the fast Gauss transform to option pricing. Working paper, Graduate School of Business, Columbia University, 2002.

[23] C. Chatfield. *The Analysis of Time Series: An Introduction.* Chapman and Hall, London, Fourth edition, 1989.

[24] R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Levy models. Internal Report 513, CMAP, Ecole Polytechnique, 2003.

[25] C.W. Cryer. The efficient solution of linear complementarity problems for tridiagonal Minkowski matrices. *ACM Transactions on Mathematical Software*, 9:199–214, 1983.

[26] T. Dangl. Investment and capacity choice under uncertain demand. *Journal of European Operational Research*, 117(3):415–428, September 1999.

[27] M.A.H. Dempster, J.P. Hutton, and D.G. Richards. LP valuation of exotic American options exploiting structure. *Journal of Computational Finance*, 2:61–84, Fall 1998.

[28] J.N. Dewynne and P. Wilmott. A note on average rate options with discrete sampling. *SIAM Journal on Applied Mathematics*, 55:267–277, 1995.

[29] Y. d'Halluin, P.A. Forsyth, and G. Labahn. A penalty method for American options with jump diffusion processes. Published on-line in Numerische Mathematik, 2003.

[30] Y. d'Halluin, P.A. Forsyth, and G. Labahn. A semi-Lagrangian approach for American Asian options under jump diffusion. Submitted to SIAM, Journal of Scientific Computing, 2003.

[31] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal. Managing capacity for telecommunications networks under uncertainty. *IEEE/ACM Transactions on Networking*, 10(4):579–588, 2002.

[32] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal. Robust numerical methods for contingent claims under jump diffusion processes. Submitted to IMA, Journal of Numerical Analysis http://www.scicom.uwaterloo.ca/~ydhallui, 2003.

[33] Y. d'Halluin, P.A. Forsyth, and K.R. Vetzal. Wireless network capacity investment. Working paper, University of Waterloo, http://www.scicom.uwaterloo.ca/~ydhallui, 2003.

[34] Y. d'Halluin, K.R. Vetzal, and G. Labahn. Asian options: A pricing overview. Working paper, University of Waterloo, 2003.

[35] A.K. Dixit and R.S. Pindyck. *Investment Under Uncertainty*. Princeton University Press, 1994.

[36] J. Douglas and T.F. Russel. Numerical methods for convection-dominated diffusion problems based on combining the method of characteritics with finite element or finite difference procedures. *SIAM Journal on Numerical Analysis*, 19:871–885, 1982.

[37] A. Duijndam and M. Schonewille. Nonuniform fast Fourier transform. *Geophysics*, 64:539–551, 1999.

[38] G.B. Durham and R. Gallant. Numerical techniques for simulated maximum likelihood estimation of stochastic differential equations. *Journal of Business and Economic Statistics*, (20):297–316, 2002.

[39] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM Journal on Scientific Computing*, 14:1368–1393, 1993.

[40] A. Egorov, H. Li, and Y. Xu. Maximum likelihood estimation of time-inhomogenous diffusions. *Journal of Econometrics*, 2003.

[41] E.M. Elliot and J.R. Ockendon. *Weak and Variational Methods for Moving Boundary Problems*. Pitman, 1982.

[42] B. Eraker, M. Johannes, and N. Polson. The impact of jumps in volatility and returns. *Journal of Finance*, 58:1269–1300, 2003.

[43] M. Falcone and R. Ferretti. Convergence analysis for a class of high-order semi-Lagrangian advection schemes. *SIAM Journal on Numerical Analysis*, 35(3):909–940, 1998.

[44] P.A. Forsyth and K.R. Vetzal. Quadratic convergence of a penalty method for valuing American options. *SIAM Journal on Scientific Computation*, 23:2096–2123, 2002.

[45] P.A. Forsyth, K.R. Vetzval, and R. Zvan. Convergence of numerical methods for valuing path-dependent option using interpolation. *Review of Derivatives Research*, 5:273–314, 2002.

[46] J. Fuhrmann and H. Langmach. Stability and existence of solutions of time-implicit

finite volume schemes for viscous nonlinear conservation laws. *Applied Numerical Mathematics*, 37:201–230, 2001.

[47] G. Kealey (Managing Consultant, Bell Nexxia). Private communication, 2001.

[48] M.B. Giles. On the stability and convergence of discretizations of initial value PDEs. *IMA Journal of Numerical Analysis*, 17:563–576, 1997.

[49] C. Gourieroux, A. Monfort, and E. Renault. Indirect inference. *Journal of Applied Econometrics*, (8):85–118, 1993.

[50] L. Greengard and J. Strain. The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12:79–94, 1991.

[51] L.P. Hansen and J. A. Scheinkman. Back to the future: Generating moment implications for continuous-time markov processes. *Econometrica*, (63):767–804, 1995.

[52] J. Hull. *Option, Futures, and Other Derivatives.* Prentice-Hall, Inc., Upper Saddle River, N.J., third edition, 1997.

[53] J. Hull and A. White. Efficient procedures for pricing European and American path-dependent options. *Journal of Derivatives*, 1 (Fall):21–31, 1993.

[54] C. Johnson. *Numerical Solutions of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, 1987.

[55] S.G. Kou. A jump-diffusion model for option pricing. *Management Science*, 48:1086–1101, 2002.

[56] J.F.B.M. Kraaijevanger, H.W.J. Lenferink, and M.N. Spijker. Stepsize restrictions for stability in the numerical solution of ordinary and partial differential equations. *Journal of Computational and Applied Mathematics*, 20:67–81, 1987.

[57] J.S. Lee and L.E. Miller. *CDMA Systems Engineering Handbook*. Artech House, 1998.

[58] H.W. Lenferink and M.N. Spijker. On the use of stability regions in the numerical analysis of initial value problems. *Mathematics of Computation*, 57:221–237, 1991.

[59] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, Basel, 1990.

[60] A. Lewis. Fear of jumps. In *Wilmott*, pages 60–67. Wiley, December 2002.

[61] WestBay Engineers Limited. What is an Erlang? http://www.erlang.com/whatis.html.

[62] M. Marcozzi. On the valuation of Asian options by variational methods. *SIAM Journal on Scientific Computing*, 24:1124–1140, 2003.

[63] A.M. Matache, T. von Petersdorff, and C. Schwab. Fast deterministic pricing of options on Lévy driven assets. RiskLab research report, ETH, Zurich, 2002.

[64] D. McAvoy. Private communications, 2002. Bell Mobility, Associate Director Access Technology Planning.

[65] R.C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 1973.

[66] R.C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Engineering*, 3:125–144, 1976.

[67] G.H. Meyer. The numerical valuation of options with underlying jumps. *Acta Mathematica Universitatis Comenianae*, 67:69–82, 1998.

[68] G.H. Meyer. On pricing American and Asian options with PDE methods. *Acta Mathematica Universitatis Comenianae*, 70:153–165, 2001.

[69] K. Parrott and N. Clarke. A parallel solution of early exercise Asian options with stochastic volatility. Proceedings of the 11th Domain Decomposition Conference, Greenwich, 1998.

[70] H. Pham. Optimal stopping of controlled jump diffusion processes: A viscosity solution approach. *Journal of Mathematical Systems, Estimation and Control*, 8:1–27, 1998.

[71] D. Pilipović. *Energy risk*. McGraw-Hill, New-York, 1997.

[72] O. Pironneau. On the transport diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik*, 38:309–332, 1982.

[73] D.M. Pooley, P.A. Forsyth, and K.R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23:241–267, 2003.

[74] D.M. Pooley, P.A. Forsyth, and K.R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23(2):241–267, April 2003.

[75] D.M. Pooley, K.R. Vetzal, and P.A. Forsyth. Convergence remedies for non-smooth payoffs in option pricing. *Journal of Computational Finance*, 6 (Summer):25–40, 2003.

[76] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In J. J. Benedetto and P. J. S. G. Ferreira, editors, *Modern Sampling Theory: Mathematics and Application*, chapter 12, pages 251–274. Birkhauser, 2001.

[77] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

[78] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numerische Mathematik*, 43:309–327, 1984.

[79] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numerische Mathematik*, 43(2):309–327, 1984.

[80] R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial-Value Problems*. Interscience Publishers, New York, 1967.

[81] S.A. Ross, R.W. Westerfield, J.F. Jaffe, and G.S. Roberts. *Corporate Finance*. McGraw-Hill Ryerson Limited, Toronto, Third Canadian edition edition, 2003.

[82] E.S. Schwartz and M. Moon. Rational pricing of internet companies revisited. Working paper, The Anderson School, UCLA, 2001.

[83] L.O. Scott. Pricing stock options in a jump-diffusion model with stochastic volatility and interest rates: Applications of Fourier inversion methods. *Mathematical Finance*, 7:413–426, 1997.

[84] D. Tavella and C. Randall. *Pricing Financial Instruments: The Finite Difference Method*. John Wiley & Sons, Inc, 2000.

[85] L. Trigeorgis. *Real Options: Managerial Flexibility and Strategy in Resource Allocation*. MIT Press, Cambridge MA, 1996.

[86] K.R. Vetzal and P.A. Forsyth. Discrete Parisian and delayed barrier options: A general numerical approach. *Advances in Futures and Options Research*, 10:1–16, 1999.

[87] J. Večeř. A new PDE approach for pricing arithmetic average Asian options. *Journal of Computational Finance*, 4(4):105–113, 2001.

[88] A.F. Ware. Fast approximate Fourier transforms for irregularly spaced data. *SIAM Review*, 40:838–856, 1998.

[89] P. Wilmott. *Paul Wilmott on Quantitative Finance*, volume 2. John Wiley and Sons Ltd, Chichester, England, 2000.

[90] H. Windcliff, P.A. Forsyth, and K.R. Vetzal. Valuation of segregated funds: Shout options with maturity extensions. *Insurance: Mathematics and Economics*, 29:1–21, 2001.

[91] H. Windcliff, P.A. Forsyth, and K.R. Vetzal. Analysis of the stability of the linear boundary condition for the Black-Scholes equation. submitted to the Journal of Computational Finance, 2003.

[92] X.L. Zhang. Numerical analysis of American option pricing in a jump-diffusion model. *Mathematics of Operations Research*, 22:668–690, 1997.

[93] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Penalty methods for American options with stochastic volatility. *Journal of Computational and Applied Mathematics*, 91:199–218, 1998.

[94] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Robust numerical methods for PDE models of Asian options. *Journal of Computational Finance*, 1 (Winter):39–78, 1998.

[95] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Discrete Asian barrier options. *Journal of Computational Finance*, 3:41–67, 1999.

[96] R. Zvan, P.A. Forsyth, and K.R. Vetzal. A finite volume approach for contingent claims valuation. *IMA Journal of Numerical Analysis*, 21(3):703–731, 2001.